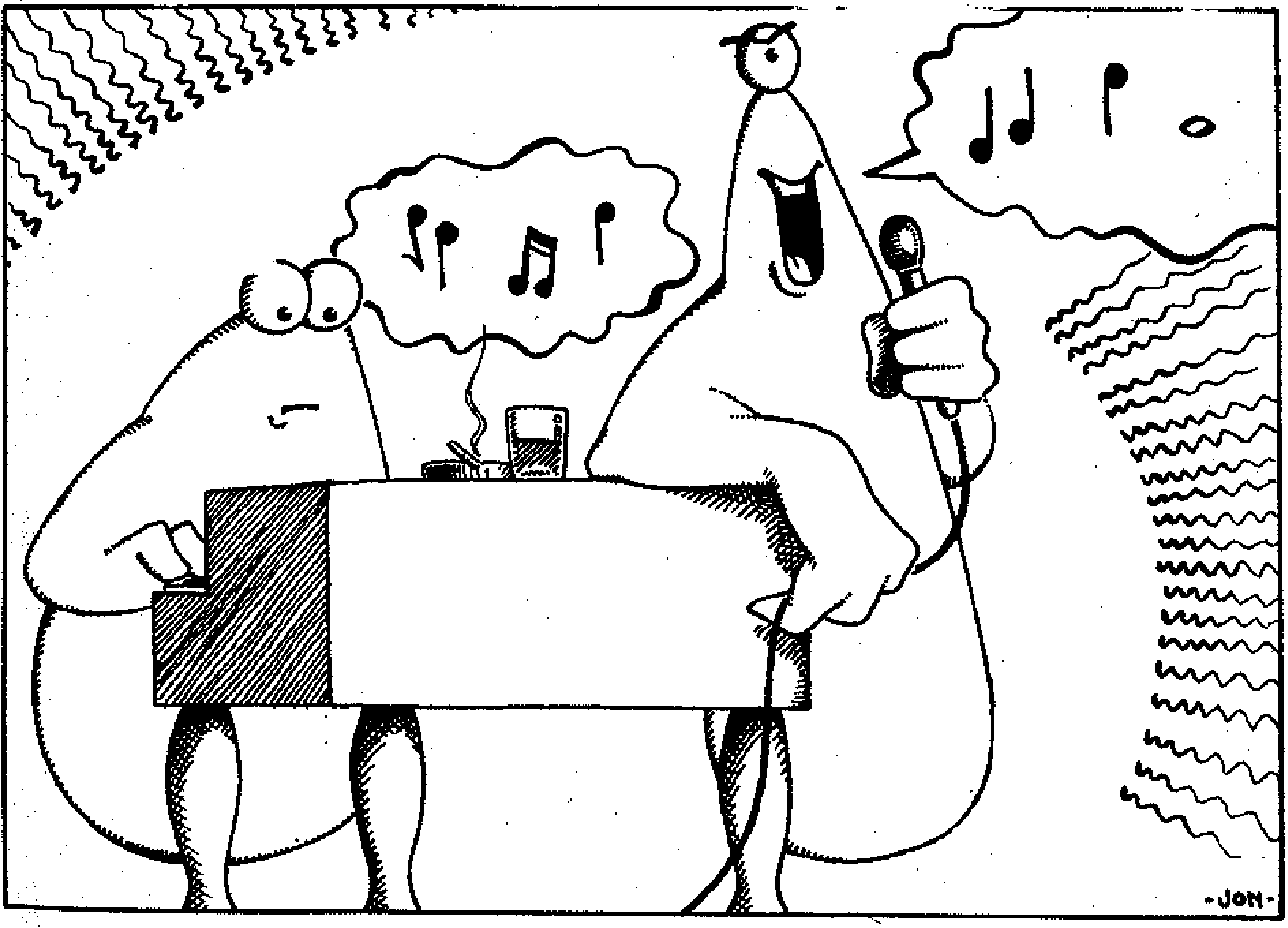
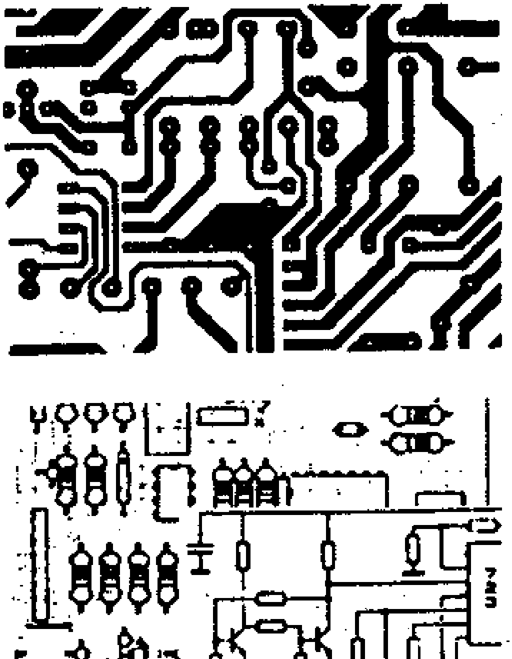
Vol 3 – No 9. May 1990.

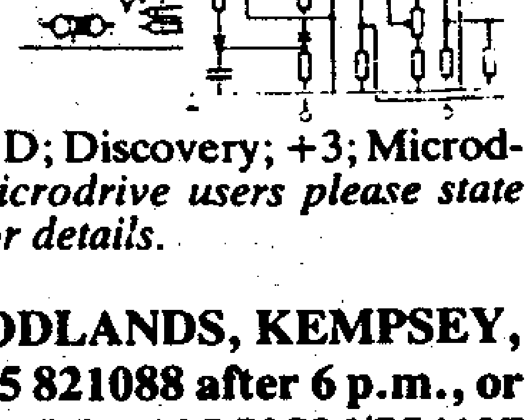
FORMAT

FOR SPECTRUM AND SAM USERS



Make Music With Your
128K Spectrum





CONTENTS

Vol.3 No.9. May 1990.

THIS MONTHS ADVERTISERS:-

ALL FORMATS SHOW	6
BETTERBYTES	Back Cover
KEMSOFT	2
LERM	20
MEL'S FUN LINE	30
P.C.G.	29
S.D.SOFTWARE	19
SHIMON YOUNG	13
STEVE'S SOFTWARE	14
TEACHER'S PET	13

FORMAT is published by FORMAT PUBLICATIONS. 34 Bourton Road, Gloucester, GL4 OLE, England. Telephone 0452-412572.

DISCIPLE , PLUS D, SAM and SAM Coupe are trade marks of MILES GORDON TECHNOLOGY plc. Lakeside, Phoenix Way, Swansea, South Wales, SA7 9EH. Telephone 0792-791100.
Printed by D.S.LITHO. Gloucester. Telephone 0452-23198.

NEWS ON 4

NEW ROM FOR SAM

MGT have announced that the new version of the SAM Coupé ROM will be sent out to owners at the end of the month. The up-grade package will contain a new ROM (V2.0) and full instructions on how to open you Coupé and fit the chip. The job is very simple and will only take about ten minutes.

All owners who purchased direct from MGT and those who have sent in their guarantee cards will receive their kits in the post. If you haven't sent off your guarantee cards please do so straight away to avoid delays.

Disc users will also receive a new DOS to match the new ROM.

At the time of going to press the latest test version of the ROM still contained a few minor bugs but things are well on target to meet MGT's deadlines.

FIRST SAM UTILITIES FROM LERM

LERM Software, Famous for its Spectrum utilities, has now produced two programs for the SAM Coupé.

The first is an assembler. LERM has taken the assembler from its Z80 TOOLKIT package and rewitten it for the Coupé with enhanced features. It contains a full-screen editor, auto line insert, line renumber, block delete and decimal, hex or binary formats for numbers. The program comes on tape at £8.99 with easy transfer to disc. A matching disassembler and other additions will be available soon at discount prices to early purchasers on the assembler.

At the same time LERM has launched a package called SAMTAPE containing two programs that allow you to use most of your favourite Spectrum programs on your SAM. It is designed for tape to disc transfer of your Speccy programs. SAMTAPE also costs £8.99 on tape.

Further details are available from LERM Software, 11 Beaconsfield Close, Whitley Bay, Tyne & Wear, NE25 9UW, Tel:- 091-2533615.

LEVEL 9 SCOOP AWARD

Scrape Ghost helped Level 9 win the Golden Chalice Award at the 1989/90 Adventure Club Awards. This is the fifth year running that Level 9 have walked of with this top award and it is a record unmatched by any other company.

The awards are voted for by members of the Adventure Club Ltd, they reflect the end users, those who play adventure games.

STEVE'S ASSEMBLER

Steve Nutting (Steve's Software) has announced a new assembler for the SAM Coupé. Called SC_ASSEMBLER, the program is written for the 256k Coupé with a version for the 512k machine to come out later this year.

Source code is compressed to allow more to fit into memory - it is claimed that you can store up to 10,000 lines of source.

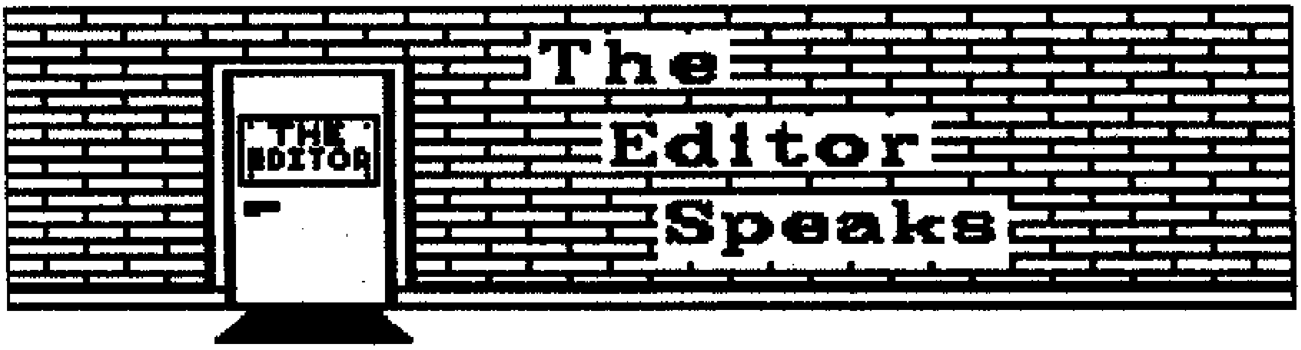
The program costs £10 and is available from Steve's Software, 7 Narrow Close, Histon, Cambridge, CB4 4XX, Tel:- 0223 235150

SD TAKEOVER

SD Software is now under new management. Nev Young (of Help Page fame) liked 'Hackers Workbench' so much he bought the company.

Funny, I seem to remember a similar story about an electric shaver.

URGENT we need your news. Clubs, Shows, New Releases, anything you think other people should know about.



The Editor Speaks

First item this month is an appeal. Our dear friends (ha ha) at the Post Office seem to have had their fun again. When the March issue of FORMAT was sent out some sixty odd orders for tapes and back-issues where dispatched at the same time. So far over twenty people have contacted me saying that their orders did not arrive. In their usual unhelpful way the Post Office fail to show much in the way of interest so it looks like more losses for FORMAT. Still, if anyone ordered items in February and didn't get them as expected in early March, PLEASE RING ME NOW so I can add you to the list. One day someone will set up in competition to the Post Office and I will be at the head of the queue.

Some of you will notice a change to the typeface in this months issue. I've splashed out on an Ink-Jet printer to speed up production of FORMAT (well anything is faster than a daisy-wheel printer). I hope you like the new print, it will enable more flexible print styles in future - once I get time to play with the control codes a bit more. Your comments, good or bad, about the change will be read with interest.

As always, I'm looking for more writers for FORMAT. I would particularly like to have a regular column in FORMAT that looks at the more 'Intellectual' type of games: Chess, Backgammon, Strategy etc. Lots of people enjoy this aspect of computing (some would object to me calling them games). As FORMAT is the only Spectrum/SAM magazine that treats its readers as if they had some intelligence a 'Mind Games' column would go down quite well. Anyone interested in taking on this task? Give me a ring.

Carol Brooksbank has asked me to make an appeal on her behalf. She needs your fonts. No not the Baptismal type, the Type type... Carol is trying to build up a library of screen fonts which she hopes to make available to readers at a later stage. So if you have designed any on-screen fonts for the Spectrum/SAM computers, you know - the 8x8 sets not the high-res sets like Qualitas, then send a copy (on tape or disc) to Carol Brooksbank, 14 Willow Grove, Tile Hill, Coventry, CV4 9EP.

The 'Small Ads' section of FORMAT has been missing for a few months because there hasn't been enough adverts coming in. Next month it will return, if you haven't had your advert published yet then send it in again. Small Ads are free to readers so why not use them? Also remember that the Small Ads section is the ideal place to publish your address if you want to contact other users locally.

Rumours have reached my ears that other magazines are trying to muscle in on the SAM. New Computer Express, Popular Computing Weekly and Your Sinclair have all started small columns devoted to the Coupé. Nice to see the opposition catching up - but they have a long way to go to catch up with FORMAT.

Finally. I will be away from Gloucester for the last week in May and the first week in June, so the Hotline will be off during that time.

Until next month,

Bob Brenchley. Editor.

THE PLACE TO BE...

ON: SATURDAY JUNE 9th – 10am-5pm
SUNDAY JUNE 10th – 10am-4pm

AT: THE NEW HALL of the ROYAL
HORTICULTURAL SOCIETY
GREYCOAT AND ELVERTON
STREETS WESTMINSTER · LONDON.

3rd *All Formats* COMPUTER FAIR

NEAREST TUBES: VICTORIA
PIMLICO
ST JAMES PK

Low Cost Stands

STAND SALES AND ADVANCE TICKETS
JOHN RIDING: 0225 · 447453
MIKE HAYES: 0457 · 875229

Admittance £3.00

BE THERE!

SHORT•SPOT

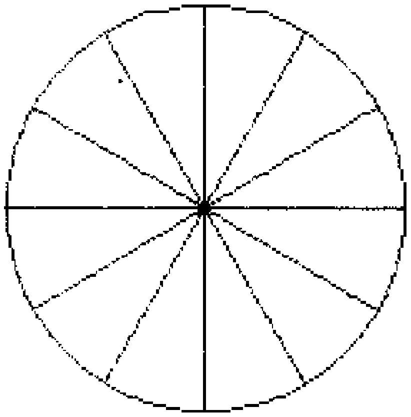
By:- John Wase.

Let's start with an appeal. Firstly to Gianni Manea of Italy. He has sent me a marvellous suite of programs for FORMAT, which do all sorts of magic things like copying discs incredibly quickly. There are 17 files on the disc run by a menu program, with all the REMs in Italian, and many of the files in machine code. There's no Basic loader or disassembly of the machine code bits, and I'm blessed if I can find out which bits go where. Gianni, it's super, but frustratingly, because it works well, we can't do much with it as it is. Please could you send just the copy program, with the code files listed with a Basic loader. Then we can take it from there. And, for everyone else reading this, please try and keep it simple, preferably a minimum number of files on disc, otherwise my task becomes impossible.

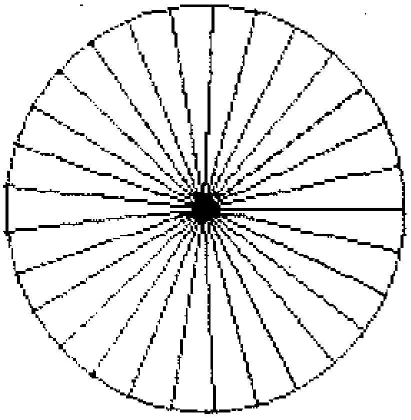
Here's a little piece from Carol. Yes, Carol Brooksbank of course, bless her and it's a real short-spot job. Carol was faced with the problem of leading a Church group some years ago - the leaders were required to cut a circle up into the number of pieces required to give each person present his or her share of the cake. In other words, segments were to be of equal areas. So, being Carol, she wrote this program:-

```
 10 REM divides circle into given numb
    er of segments
 20 REM R=radius, S=number of segments
 30 LET R=80
 40 INPUT "NUMBER OF SEGMENTS REQUIRED
    ? ";S
 50 LET N=S/2
 60 PLOT 120,90
 70 CIRCLE 120,90,R
 80 FOR X=0 TO N
 90 LET A=PI/N*X
100 PLOT 120,90
110 DRAW R*COS A,R*SIN A
120 DRAW -R*COS A,-R*SIN A
130 DRAW R*COS A,-R*SIN A
140 DRAW -R*COS A,R*SIN A
150 NEXT X
```

So far so good. You get pictures like those below when you ask for an integer number of segments.



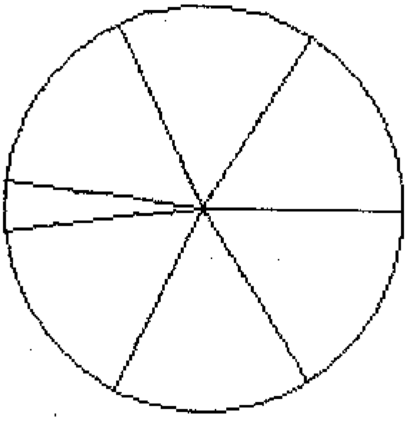
Number of Segments = 12



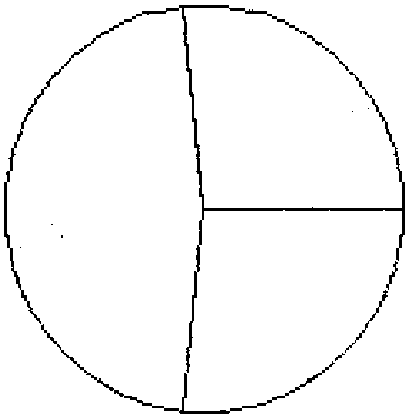
Number of Segments = 29

But, Carol, being Carol, wondered what would happen if she stuck in some decimals. So she experimented. She found that if she took a decimal number, N.D and N was an even number, then the circle was divided into N+1 segments, the N all being the same size and the extra one being .D as wide as the others. And if N was an odd number, then the circle was divided into N segments, but one of them was 1.D as wide as the rest.

Here's some examples.



Number of Segments = 6.25



Number of Segments = 3.75

Funny, isn't it. Perhaps you'd like to think about it; maybe it will stimulate you into some more curious bits and pieces.

Whilst we're on about curious bits and pieces, I've a couple of little items here from our correspondent Rusty Atkins of Reading. Like me, Rusty gets niggled by the DISCiPLE / PLUS D discs, which are not named. (Only on the label which I've forgotten to stick on). He finds that the solution is in the "+SYS" file. Although this is 10 letters long, the system only needs to find the four letters "+SYS" to boot (or just "SYS" on the DISCiPLE). The remaining six letters can be used as an ident. Take a disc, and simply type a command like:-

"ERASE D1"+SYS" TO "+SYS A-BAK"

and the deed is done. Make sure a system file is the first one on the disc and you can't go wrong. Or can't you?

You see, there is still the problem of finding the file you want to load. Now, like me, Rusty is perpetually broke, and therefore has the habit of cramming as many files as he can on a disc (like 80). The program "catprog" prints out an 80 program list in eight letter form on one screen, so that you can flick through it quickly and find if the program you want is there. Good, eh? There's just one problem, though. When you've found the program, you want to load it with "LOAD Pn" where n is the number of the program. Don't you. And, of course, on the short cat command, there's no number. So Rusty's program adds that, too. Fantastic. Here's the listing:-

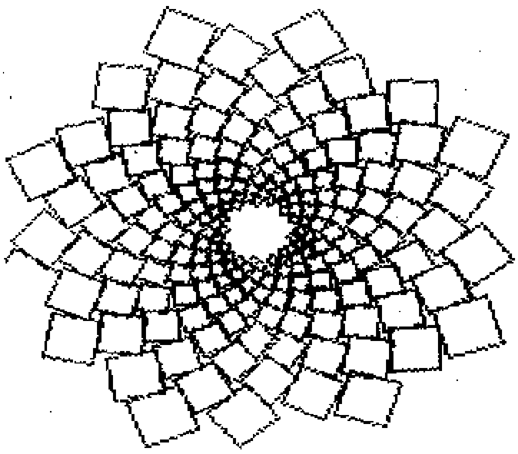
```
 10 REM ******CATPROG*****
 20 REM ***Rusty Atkins***
 30 REM
 40 REM if error message"Prog already
    exists" is given (due to using BRE
    AK) use RUN  405
 50 REM Acknowledgemt to Andrew Brown
    for some lines pinched from SECTOR
     MAP (FORMAT 2/6).
 60 PRINT AT 21,0;"       Short wait p
    lease        ": REM 7 @ 7 spaces
 70 LET C=30000: FOR T= 0 TO 3: FOR S=
     1 TO 10: LOAD @1,T,S,C
 80 LET C=C+512: NEXT S: NEXT T
100 LET A=0: LET B=0: DIM A(80)
110 CLS : LET X=0: FOR P=0 TO 79
120 LET D=30000+P*256: IF PEEK D=0 THE
    N GOTO 190
130 DIM N$(10)
140 FOR I= 1 TO 10: LET N$(I)=CHR$ PEE
    K (D+I): NEXT I
150 IF INT (X/2)=X/2 THEN BRIGHT 1
155 IF INT (X/2)<>X/2 THEN BRIGHT 0
160 PRINT AT B,A;N$(1 TO 8): LET X=X+1
170 LET A=A+8: IF A=32 THEN LET A=0: L
    ET B=B+1
180 LET A(P+1)=P+1
190 NEXT P
200 SAVE !"DIR1"SCREEN$
210 DIM B(80): LET R=1: FOR N=1 TO 80
220 IF A(N)<>0 THEN LET B(R)=A(N): LET
     R=R+1
230 NEXT N
240 LET A=0: LET B=-1: LET R=0
250 GOSUB 300: GOSUB 300: GOSUB 300: G
    OSUB 300
260 LET A=A+1: LET B=-1: GOTO 250
300 LET B=B+8: IF B>31 THEN LET B=0
310 LET R=R+1: IF R>80 THEN GOTO 400
```

```
320 IF R>=10 THEN GOTO 340
330 IF B(R)<>0 THEN PRINT  INVERSE 1;A
    T A,B-1;"0";B(R); INVERSE 0: GOTO
    350
340 IF B(R)<>0 THEN PRINT  INVERSE 1;A
    T A,B-1;B(R); INVERSE 0
350 RETURN
400 SAVE ! "DIR2"SCREEN$
405 LET Y=1
410 PRINT AT 21,0; "T=TOGGLE,N=NEW CAT
    ,L=LOAD,E=END"
420 IF INKEY$<>"" THEN GOTO 420
430 IF INKEY$="" THEN GOTO 430
440 LET A$=INKEY$
450 IF A$="E" OR A$="e" THEN ERASE !"D
    IR1": ERASE !"DIR2": STOP
460 IF A$="T" AND Y=1 OR A$="t" AND Y=
    1 THEN LOAD !"DIR1"SCREEN$ : LET Y
    =2: GOTO 410
470 IF A$="T" AND Y=2 OR A$="t" AND Y=
    2 THEN LOAD !"DIR2"SCREEN$ : LET Y
    =1: GOTO 410
490 IF A$="N" OR A$="n" THEN ERASE !"D
    IR1": ERASE !"DIR2": GOTO 60
500 IF A$="L" OR A$="l" THEN ERASE !"D
    IR1": ERASE !"DIR2": GOTO 515
510 IF A$<>"N" AND A$<>"n" AND A$<>"T"
     AND A$<>"t" AND A$<>"L" AND A$<>"
    l" THEN GOTO 420
515 LET B$=""
520 PRINT AT 21,0;"ENTER PROG. NO. ONL
    Y (2 figs)   "
530 FOR N= 1 TO 2
540 IF INKEY$<>"" THEN GOTO 540
550 IF INKEY$= "" THEN GOTO 550
560 IF CODE INKEY$<48 OR CODE INKEY$>5
    7 THEN GOTO 520
570 LET B$=B$+INKEY$: NEXT N
580 PRINT AT 21,0; "Loading prog. ";VA
    L B$;"                ": REM 16 sp
    aces
590 LOAD p(VAL B$)
```

There's just a couple of problems that Rusty has. Firstly, the program hangs if there's no disc in the drive (just insert a disc to cure that one). And secondly, after it has loaded the program, it prints out an error message, like "Nonsense in GDOS" if you load a program that doesn't have and auto-run line number, irritating but your program will have loaded OK. Apart from this, there are one or two rough edges as well, but they're not serious. Rusty's pretty frustrated by this one, though. Anyone got the solutions. Answers on a postcard, please.......

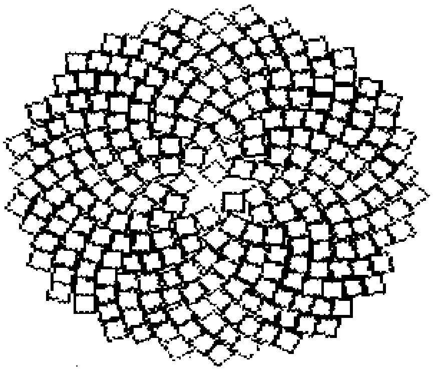
Ettrick Thompson of Aldeburgh has written to me several times recently. He is familiar with "Mathographics" which he mentions was first published in 1987 at a cost of £9.95. Ettrick points out that whilst it is a fascinating book, it is also exasperating because firstly you haven't a hope in hell of reproducing many of the illustrations on the Spectrum screen, and secondly because many of the programming hints are very sketchy indeed.

Ettrick says that the previous illustration in Short Spot was a "False Daisy": he prefers the "True Daisy" and "Sunflower", both of which use c=(1 + SQR 5)/2 =1.618034, the so called "golden ratio", the only choice which gives even packing. Here are the sunflower and daisy programs:-

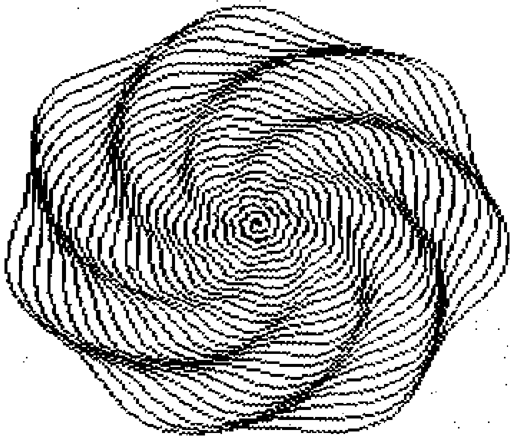


```
  5 REM Mathographics; Sunflower
 10 INPUT "Centre Size:";R'"Rate of in
    crease:";H
 20 PRINT R;TAB 24;H
 30 FOR A=0 TO 1000 STEP 2*PI/1.618034
 40 LET R=H*R
 50 LET D=R: LET E=R/5
 60 GOSUB 200
 70 NEXT A
 80 STOP
 90 :
 95 REM Mathographics; Daisy
100 FOR A=0 TO 1000 STEP 2*PI/1.618034
110 LET R=2*SQR A
120 LET D=R: LET E=5
130 GOSUB 200
140 NEXT A
```

```
150 STOP
160 :
200 LET C=COS A: LET S=SIN A
210 LET U=E*C: LET V=E*S
220 PLOT D*C+128,D*S+88
230 DRAW U+V,V-U
240 DRAW U-V,V+U
250 DRAW -U-V,-V+U
260 DRAW -U+V,-V-U
270 RETURN
```

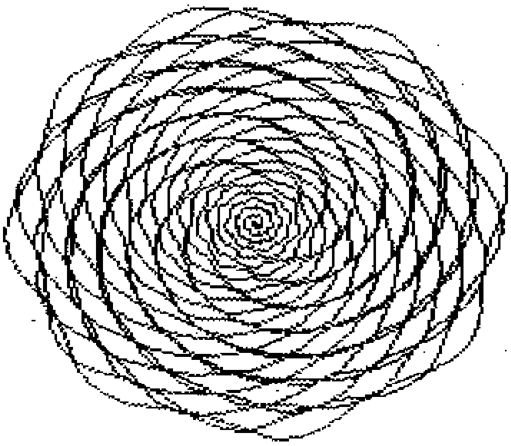


Ettrick also says that the "Rose curve with a twist" is shown in Mathographics, but is so vague in the programming direction, it took a long while to sort out. Here it is.



```
  5 REM Rose Curve with a Twist
 10 PLOT 128,88
 20 FOR A=0.1 TO 170.5 STEP 0.1
 30 LET R=A*(.5+.025*SIN (7.1*A))
 40 DRAW 128+R*COS A-PEEK 23677,88+R*
    SIN A-PEEK 23678
 50 NEXT A
```

Now change the 7.1 in line 30 to 7.7 and you get the pattern shown at the top of the next column.



Now a confession. Ettrick's latest letter has gone walkabout. My fault entirely. Please, Ettrick, could you send the one about integer arithmetic again. Thanks.

Finally, here's a bit of entertainment from our old friend L.G.Baumann of Cowies Hill, Pinetown, South Africa. Can you see how this works?

```
  1 REM ****HOWWORK?*****
  2 REM ** L.G.BAUMANN **
 10 LET D$=" HOW DOES THIS WORK?"
 20 LET B$="": LET A=19: LET B=5
 30 FOR N=1 TO 20
 40  LET  B$=B$+D$(N)+CHR$  22+  CHR$
    A+CHR$ B
 50 LET A=A-1: LET B=B+1
 60 NEXT N
 70 LET A$=B$
 80 PRINT B$: PAUSE 70
100 CLS : FOR N=2 TO 74 STEP 4
110   PRINT  B$(N  TO  N+3):  BEEP
    .2,(N+2/2)
120 NEXT N
130 PRINT AT 0,0;D$
140 PAUSE 0: GOTO 100
```

Mr Baumann mentions that it is rare for anyone to need to use instructions embedded in a string, but, nevertheless, the Spectrum makes provision for this as explained in the Manual.

Well, how DOES it work?

Send any contributions for Short Spot to:- John Wase, Green Leys Cottage, Bishampton, Pershore, Worcs, WR10 2LX.

MUSICAL INTERFACING

By:- Frank Kirby.

Now that the SAM Coupé is here I look forward to getting it hooked up to my existing MIDI system. In the meantime if anyone is interested in using their 128K or +2 to generate music, other than with the simple BEEP command, then read on.

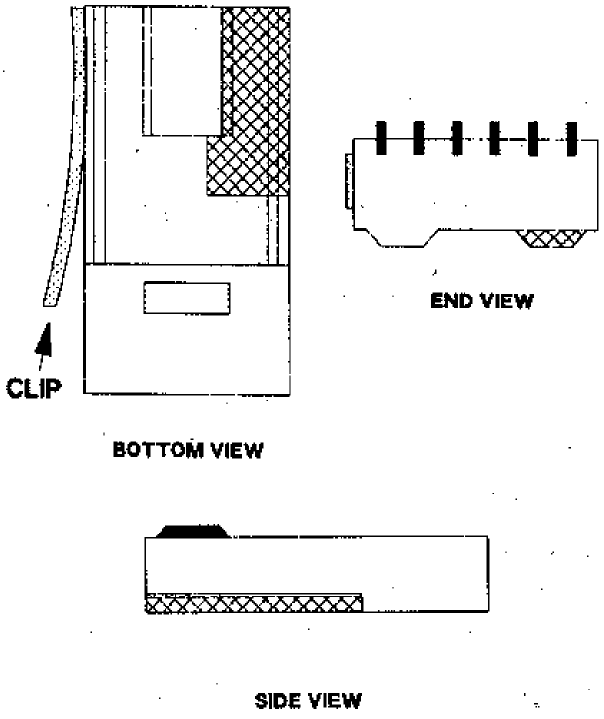
The system I use at the moment is a 128K Spectrum and a Yamaha PSS480 plus a Casio HT700. I considered buying a proper MIDI interface and software for about £50.00 but decided against it. The main reason was the fact that it would only address two channels at any one time. The alterative method I have set up cost me just £3.50. It is only able to send MIDI data out and not receive it, but this should be sufficient for most budding musicians.

The technique is purely in BASIC and can address up to eight channels at any one time (although in some instances I can use all sixteen if necessary). The articles in FORMAT by Ray Elder were quite fascinating and yet I can imagine that unless you are a MIDI 'Buff' you may find it hard slog after a few pages of reading.

However if you wish to connect your 128K/+2 to a MIDI keyboard such as the Yamaha PSS480/580/680/780 or Casio HT series then this article should get you started. If you are used to sending escape codes to a printer then this couldn't be easier. If you can already use the internal sound chip of the 128K with PLAY "gedegede" etc. then you are almost there.

First of all you need to make up a DK Mark 1 MIDI lead to suit the Spectrum. This consists of a BT 6way plug, a suitable length of cable and a 45° 5 pin DIN plug. The BT plug needs a little hacking to get it to fit the Spectrum (see Fig.1) and this is best done, carefully, with a sharp knife

Fig.1 - Modifing the BT Plug



CUT CAREFULLY THE SHADED AREAS

Once modified, you need to clamp the BT plug over the wires in the lead and solder the DIN plug to the other end (see Fig.2 a & b for connections).

Fig.2a - BT Plug Connections.

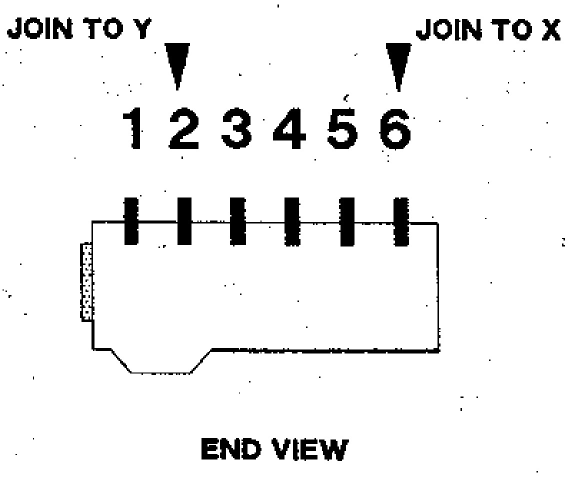
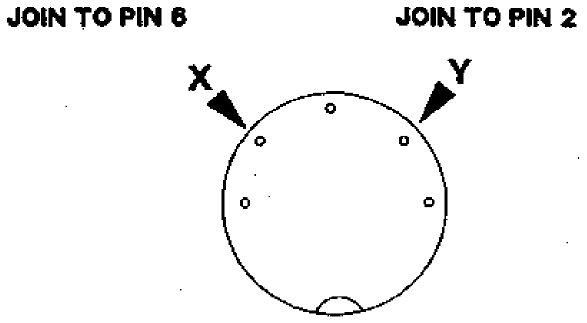


Fig.2b - DIN Plug Connections.



5 PIN DIN PLUG (45°)

FRONT VIEW

Now you are ready to go. The following examples work on a Yamaha PSS.480. They should also work on the 580/680/780.

1) The computer and keyboard should both be switched off.
2) Plug the 5 pin DIN into the MIDI IN on the instrument.
3) Plug the BT plug into the MIDI/RS232 socket on the 128/+2/+3.
4) Switch on the computer and the instrument.
5) Switch the PSS.480 to Auto accompaniment OFF and to sound source mode (99).[See manual]

The rest of the article assumes that you can read music or can translate those tadpoles and bubbles on sheet music into the strings needed to PLAY.

Assume the sixteen channels on the instrument are 1-16 as opposed to 0-15. Without going into too much detail at this stage we need to set the CHANNELS on which we are going to PLAY our composition.

```
10 PLAY "Z192Z13": REM  THIS  SETS
   CHANNEL  1  TO THE VOICE 13 (ROCK
   GUITAR).
20 PLAY "Y1T70N4g1a3g6e5D3D6b5C3C6g"
30 REM Y1.....CHANNEL NUMBER
       T70....TEMPO
       N......NOTE LENGTH
```

It's not very inspiring at this stage as we have only played one line of music on one CHANNEL. Let's add more lines to make the notes into CHORDS.

Edit line 20 and add a comma followed by 3 more strings.

```
20 PLAY "Y1T70N4g1a3g6e5D3D6b5C3C6g"
   ,"Y1N4e1f3e6c4e1f3e6c5g3g6g5g3g6e
   ","Y1O4N4C1C3C6g5F3F6F5E3E6C","Y1
   N5&&&&&&O4b3b6D5g3g6&"
```

It sounds a little better. The CHORDS are being PLAYed on CHANNEL 1.

Let's add another voice. Edit line 10.

```
10 PLAY "Z192Z13Z193Z20":REM    SETS
   VOICE 20 TO CHANNEL 2.
```

Edit line 20 and add a comma followed by 3 more strings.

```
20 PLAY "Y1T70N4g1a3g6e5D3D6b5C3C6g"
   ,'Y1N4e1f3e6c4e1f3e6c5g3g6g5g3g6e
   ","Y1O4N4C1C3C6g5F3F6F5E3E6C","Y1
   N5&&&&&&O4b3b6D5g3g6&","Y2N5&&C&&
   C&&g&&C","Y2N5&&g&&g&&&&&&","Y2N5
   &&e&&e&&&&&&"
```

This PLAYs the notes on CHANNEL 2 as well as those specified on CHANNEL I. Finally you can add a BASS line by setting up CHANNEL 3.

Edit line 10...

```
10 PLAY "Z192Z13Z193Z20Z194Z15":REM
   SETS VOICE 15 TO CHANNEL 3.
```

Edit line 20 again and add a comma followed by one more string.

```
20 PLAY "Y1T70N4g1a3g6e5D3D6b5C3C6g"
   ,'Y1N4e1f3e6c4e1f3e6c5g3g6g5g3g6e
   ","Y1O4N4C1C3C6g5F3F6F5E3E6C","Y1
   N5&&&&&&O4b3b6D5g3g6&","Y2N5&&C&&
   C&&g&&C","Y2N5&&g&&g&&&&&&","Y2N5
   &&e&&e&&&&&&","Y3O3N6_6c6_6c6_6G6
   _6c"
```

The numbers in line 10 which I will refer to as Z codes select various parameters known as STATUS and DATA bytes to be sent.

```
Z192Z13 ... CHANNEL 1 VOICE 13
```

```
Z193Z20 ... CHANNEL 2 VOICE 20
Z194Z15 ... CHANNEL 3 VOICE 15
```

You can assign any VOICE to any CHANNEL using this technique. Z192 to Z206 equals CHANNELS 1 to 15. Z207 is CHANNEL 16 and is usually reserved for RHYTHM DATA or MUSICAL STYLE depending on the MODE you are operating in.

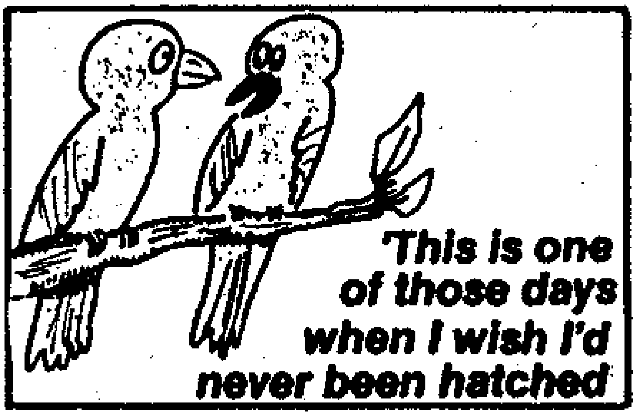
You can assign any note to be played on any CHANNEL with:-

```
PLAY "Y(1-15)..NOTE DATA..","Y(1-15)..
NOTE DATA.." etc.
```

If you already have your own compositions which you use to PLAY with the 3 channel sound chip through your TV, you just have to add a line at the beginning to ASSIGN VOICES to CHANNELS and then edit each line of your composition to include the "Y code" to specify the CHANNEL to use.

At a later stage I will show you how to include RHYTHM patterns within strings, set vibrato, portamento and sustain switches.

I have almost finished a DRUM PATTERN EDITOR which allows you to build up your drum sequences in steps, edit them and PLAY them back. The next project is a SYSTEM EXCLUSIVE DUMP to edit the VOICE BANKS. All these are in BASIC so no knowledge of Z80 or MIDI protocol is necessary.



STEVE'S SOFTWARE

SC_ASSEMBLERis a powerful Editor Assembler specially written for the SAM COUPE 256K Computer with 1 or 2 Disc Dives. SC_Assembler allows you to create upto 20K Machine code programs with the help of a friendly 64 column editor system.

The editor will except source like no other Spectrum Assembler, First there is no need to type spaces between opcodes and labels no need to tab and field source to certain parts of a line.

For example you could enter a source line like :-

```
10label:ldhl,16384  or 10 La BE  l  :L  dH  l, 183 8 4
```

When you press Enter, the line is reprinted like this :-

```
10      label: LD   HL,16384
```

Not only is it quicker and easier to enter source, SC_ASSEMBLER will compress the source so it takes up half the space compared to other Spectrum Assemblers,with 96K of source space availiable it's possible to store around 10,000 lines of source code.

The editor allows you to edit the whole screen of source, with the Function keys you can bring more lines of source to the screen in so many different ways allowing for faster editing.

If you happen to type somthing wrong then helpful worded error messages are displayed, you don't get messages like ERROR 02.

Assembling source on Pass 1 produces helpful info on where your code starts,ends,length and number of labels used before even going onto pass 2.

SC_ASSEMBLER with the built in Dissasembler will recognize the Full set of 798 opcodes, including the 102 Undocumented ones.

SC_ASSEMBLER is a massive 40K code program designed not to use any of the SAM ROM routines, to ensure this Quality product will be compactable with different ROMs that may appear in the Future

SC_ASSEMBLER costs £10, available from :- STEVE'S SOFTWARE
7 NARROW CLOSE, HISTON, CAMBRIDGE, CB4 4XX.
Please make cheques payable to MR S.J.NUTTING

A 512K version Assembler will be availiable around September. This will surport the extra 256K memory, useful SAM source files built in Monitor and a very large DTP produced Manual with lots of Technical information for Advanced Users, at a cost of £15.

If you purchase the 256K version now you can upgrade to the 512K version for £5, so you don't lose out buying now.

SAM's
SMALL IS BEAUTIFUL

By:- Carol Brooksbank.

A number of people have been kind enough to tell me that they find my 'Small is Beautiful' screendump and filofax page printing programs for the Spectrum useful, so I thought that SAM users might find them handy too. So here are SAM versions of the listings.

Briefly, for those who haven't met it before, 'Small is Beautiful' is a machine code program for small screendumps - down to 1 printed dot per screen pixel. This allows you to use the whole screen to get plenty of detail into your design, but still have a tiny printout. The SAM program source code is Listing 1.

LISTING 1.

```
  5 *SCREEN   ON
    *LIST     ON
    *PRINTER  ON
    *LLIST    ON
 10        ORG  60000
 20        LD   HL,(SCRNPOS)
           LD   B,32
 30 LOOP5:PUSH BC
           PUSH HL
           LD   B,8
 40        LD   DE,BYTES
    LOOP1:LD   A,(HL)
           LD   (DE),A
           INC  H
           INC  DE
           DJNZ LOOP1
 50        LD   B,8
    LOOP4:PUSH BC
           LD   HL,BYTES
           XOR  A
           LD   D,A
           LD   B,8
 60 LOOP2:RL   (HL)
           RL   D
           INC  HL
           DJNZ LOOP2
 70        LD   A,(DOTS)
           LD   B,A
    LOOP3:LD   A,D
           CALL 385
           DJNZ LOOP3
           POP  BC
          DJNZ LOOP4
 80        POP  HL
           POP  BC
           INC  HL
           DJNZ LOOP5
           RET
 90 BYTES:DEFS 8
    SCRNPOS:
           DEFS 2
    DOTS :DEFS 1
100 NXDOWN:
           LD   HL,(SCRNPOS)
           RR   H
           RR   H
           RR   H
           LD   BC,32
           ADD  HL,BC
           RL   H
           RL   H
           RL   H
           LD   (SCRNPOS),HL
           RET
```

SAM screen mode 1 is used for this program. It was developed for the Spectrum, so the machine code needs very little modification if we use the Spectrum emulating mode. Since the final result is to be an unshaded dump in monochrome, there is very little point in rewriting it to use SAM's mode 4, where, anyway, we should have to compromise about which colours were to be printed white and which black.

In fact, the only change from the Spectrum version in the machine code is in the sending of the bytes to the printer. For the Spectrum, we had to print to stream 3. This is not necessary with SAM, since there is a ROM routine at address 385 which sends the byte in the A register to the printer. So we save a hefty 3 bytes!

I have not included a detailed explanation of what the machine code is doing. This was fully covered in the original article in FORMAT vol 2, no. 5, December 1988, and back numbers are available for newcomers to FORMAT who would like to know more about the technicalities of the program.

The BASIC is very different, however. Listing 2 is the screendump BASIC. The main SAM screen file (screen 1), is at address 245760. If you want to check that, try:- POKE 245760,255. A short line will be drawn at the top left hand corner of the screen.

LISTING 2.

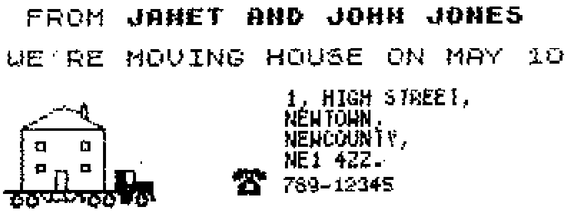
```
  10 MODE 1
  20 CLEAR 59999
  30 LOAD "dumpcode" CODE 60000
  40 OPEN #5,"b"
  50 INPUT "GRAPHICS MODE?";graphmode:
      INPUT "DOTS?";dots: INPUT "MARGI
     N?";margin
  60 INPUT "NAME OF SCREEN TO PRINT?";
     S$
  70 POKE 60064,dots
  80 INPUT "NUMBER OF COPIES?";N
  90 LOAD S$ CODE 245760
 100 LET P$=MEM$(245760 TO 252672)
 110 POKE 49152,P$
 120 PRINT #5, CHR$ 27;"1";CHR$ margin
     ;CHR$ 27;"3";CHR$ 22;
 130 FOR R=1 TO N
 140 DPOKE 60062,49152
 150 FOR Q=1 TO 24
 160 GO SUB 230
 170 CALL 60065
 180 NEXT Q
 190 PRINT #5, CHR$ 12
 200 NEXT R
 210 CLOSE #5
 220 STOP
 230 PRINT #5, CHR$ 27;"*";CHR$ graphm
     ode;CHR$ 0;CHR$ dots;
 240 CALL 60000
 250 PRINT #5,CHR$ 10
 260 RETURN
```

Those familiar with machine code will know that this is beyond a two-byte address, so on the face of it, it seems that we shall have to do convoluted things to point to the screen file and read its bytes.

Fortunately, that is unnecessary, thanks to a very useful SAM BASIC command, MEM$. This lets you copy a whole block of memory into a string, and then poke the string into memory somewhere else. So, in lines 100 to 110, we copy the screen file into the string P$, then poke it into memory starting at 49152, from where we can access it very easily by using the DPOKE command in line 140 to poke the correct low and high bytes into the two bytes at 60062-3.

The number of screen lines printed is controlled from BASIC, so that you need not print the whole screen. This is invaluable if, say, you are printing on peel-off labels (see Fig.1). The labels I use need only 15 screen lines, so by changing line 150 in listing 2 to read '150 FOR Q=1 TO 15' I can match the label size exactly.

Fig.1



FROM JANET AND JOHN JONES

WE'RE MOVING HOUSE ON MAY 10

1, HIGH STREET,
NEWTOWN.
NEWCOUNTY,
NE1 4ZZ.
☎ 789-12345

Line 40 opens a stream for sending binary data to the printer, which is essential for the printer control codes.

The program as written assumes that your printer supports Epson-type graphics ESC "*". The precise graphics mode is selected by the figure which is sent to the printer immediately after "*" - our variable 'graphmode'. The variable 'dots' governs the number of times each byte is printed. By changing the combinations of graphmode and dots, you can get all sorts of different printouts. Graphmode 4, dots 1 and graphmode 3, dots 3, give almost perfect proportions on my printer. Other combinations give varying degrees of distortion.

If your printer does not have ESC "*", you can replace "*";CHR$ graphmode; in line 230 with "K","L","Y" or "Z" - whatever your printer uses - and omit the "graphmode" INPUT from line 50.

'Margin' governs the number of spaces to the left of the printout.

For those who have no access to an assembler (I had to write the source code on the Spectrum), I have included a code poker. (Listing 3).

LISTING 3.

```
  10 CLEAR 59999
  20 FOR X=60000 TO 60087
  30 READ A
  40 POKE X,A
  50 NEXT X
  60 DATA 42,158,234,6,32,197,229,6
  70 DATA 8,17,150,234,126,18,36,19
  80 DATA 16,250,6,8,197,33,150,234
  90 DATA 175,87,6,8,203,22,203,18
 100 DATA 35,16,249,58,160,234,71,122
 110 DATA 205,129,1,16,250,193,16,228
 120 DATA 225,193,35,16,208,201,0,0
 130 DATA 0,0,0,0,0,0,0,216
 140 DATA 3,42,158,234,203,28,203,28
 150 DATA 203,28,1,32,0,9,203,20
 160 DATA 203,20,203,20,34,158,234,201
 170 SAVE "dumpcode" CODE 60000,88
```

Turning now to the filofax pages program, I did not expect to make many changes to the original because the Spectrum program was written in BETA BASIC, which is very like SAM BASIC. However, I discovered that there are some differences in the screen proportions, and the co-ordinates used in the Spectrum program produced on SAM a page which was too short and had the punch hole markings in the wrong places. So the co-ordinates used here are different from the Spectrum ones.

Listing 4 produces and saves three separate screens which are printed one after the other to produce a Filofax page. This listing produces a blank page. Again, we work in mode 1 because the printout will use the 'dumpcode' machine code, though with somewhat different BASIC.

LISTING 4.

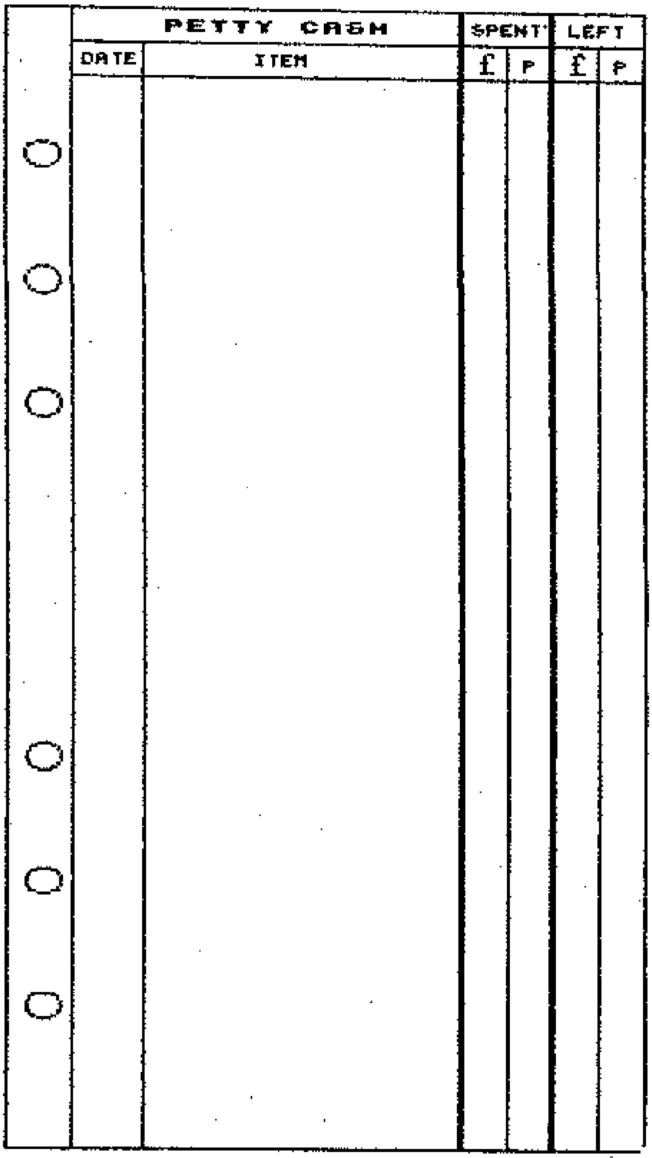
```
  10 MODE 1
  20 PAPER 7: PEN 0: CLS
  30 PLOT 0,0
  40 DRAW TO 0,175
  50 DRAW TO 224,175
  60 DRAW TO 224,0
  70 CIRCLE 13,107,6
  80 CIRCLE 13,49,6
  90 SAVE "FAX1" SCREEN$
 100 CLS
 110 PLOT 0,0
 120 DRAW TO 0,175
 130 PLOT 224,0
 140 DRAW TO 224,175
 150 CIRCLE 13,168,6
 160 CIRCLE 13,6,6
 170 SAVE "FAX2" SCREEN$
 180 CLS
 190 PLOT 0,0
 200 DRAW TO 0,175
 210 PLOT 0,0
 220 DRAW TO 224,0
 230 DRAW TO 224,175
 240 CIRCLE 13;124,6
 250 CIRCLE 13,66,6
 260 SAVE "FAX3" SCREEN$
```

In the Spectrum program, I suggested using BETA BASIC for inserting type and lines etc. onto the filofax page. SAM BASIC's CSIZE is not nearly so versatile as BETA BASIC's, however, and will not produce small characters. And since all SAM owners have the FLASH graphics program, it seems sensible to load the three screens into FLASH for finishing. Working in mode 1, you can take advantage of the grid overlay to help to ensure that vertical lines which run through more than one screen are all in the same place and will line up in the printout. You can flip to the other screen modes in FLASH if you want to take advantage of some of the other facilities like insert, but you must save to disc or tape from mode 1. Fig.2 is a finished page.

FLASH seems strangely unwilling to load SAM mode 1 screens saved as SCREEN$. If you have trouble, try replacing the SCREEN$ in lines 90, 170 and 260 of listing 4 with:- SAVE ... CODE 245760,6912. FLASH will load those quite happily in mode 1.

In the printing program, Listing 5, SAM users have a great advantage over Spectrum filofax printers, in that all three screens can be held in memory at once. We don't even have to display them on the current screen while printing. They can be held safely on screens 2,3 and 4, and their bytes copied down to 49152 for printing. The addresses in lines 170, 190 and 210 are the addresses of those screens only if they have been opened as mode 1 screens. The screen mode is the figure after the screen number in lines 30-50. Each screen is called up in turn by SCREEN, and when the graphics are loaded, we return to screen 1 for prompts, inputs and so on.

Fig.2



PETTY CASH		SPENT		LEFT	
DATE	ITEM	£	P	£	P

Graphmode 2, dots 2 will produce a printout of exactly the right size. In line 320, a printout of only the top 22 lines of the screen is specified - the designs do not go down into the lower 'input' lines.

The screens are not closed at the end of the program, because it is possible to get 2 filofax pages on 1 sheet of paper. If you change margin to 40 in line 140 and put the paper through again, GOTO 130 will print a second run alongside the first.

The Spectrum original of this program is in FORMAT Vol. 2, no. 8, April 1989.

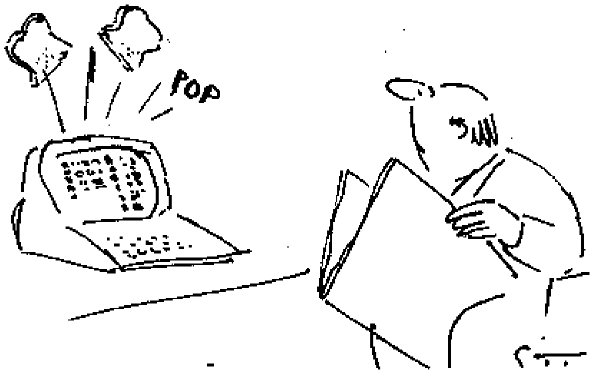
Finally, after using either the screendump or filofax printing programs you will need to reset your printer before doing anything else, because the printer is left with a line feed of $^{22}/_{216}$" spacing.

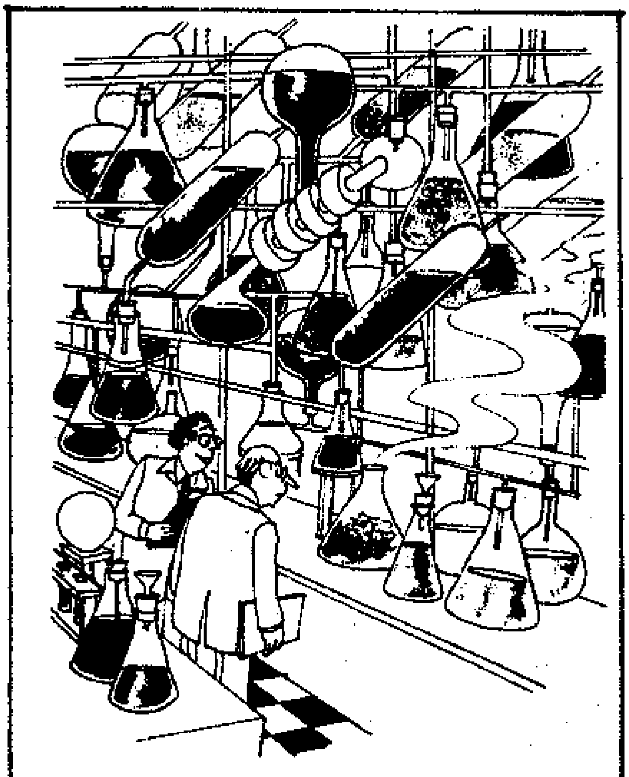
So there, for SAM users, are two Spectrum originals converted. I hope you will find them useful, and perhaps they will suggest to you ways in which you might convert other programs you have come across, to run on SAM. With SAM's splendid and versatile BASIC, most programs will run faster and/or more efficiently if you convert them carefully.

LISTING 5.

```
 10 CLEAR 59999
 20 LOAD "dumpcode" CODE 60000
 30 OPEN SCREEN 2,1
 40 OPEN SCREEN 3,1
 50 OPEN SCREEN 4,1
 60 SCREEN 2
 70 LOAD "FAX1" CODE 180224
 80 SCREEN 3
 90 LOAD "FAX2" CODE 147456
100 SCREEN 4
110 LOAD "FAX3" CODE 114688
120 SCREEN 1
130 OPEN #5,"b"
140 LET graphmode =2,dots=2,margin=0
150 INPUT "NUMBER OF COPIES?";N
160 FOR R=1 TO N
170 LET P$=MEM$(180224 TO 187136)
180 printit
190 LET P$=MEM$(147456 TO 154368)
200 printit
210 LET P$=MEM$(114688 TO 121600)
220 printit
230 PRINT #5, CHR$ 12
240 NEXT R
250 CLOSE #5
260 STOP
270 DEF PROC printit
280 POKE 60064,dots
290 POKE 49152,P$
```

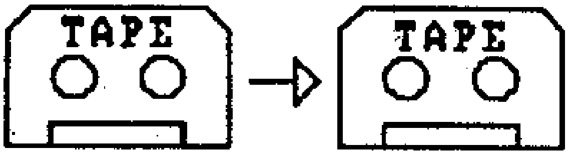
```
300 PRINT #5, CHR$ 27;"1";CHR$ margin
    ;CHR$ 27;"3";CHR$ 22;
310 DPOKE 60062,49152
320 FOR Q=1 TO 22
330 GO SUB printline
340 CALL 60065
350 NEXT Q
360 END PROC
370 LABEL printline
380 PRINT #5, CHR$ 27;"*";CHR$ graphm
    ode;CHR$ 0;CHR$ dots;
390 CALL 60000
400 PRINT #5,CHR$ 10
410 RETURN
```





No, that's the coffee...

SPECTRUM-SAM SOFTWARE by LERM. WITH FULL MONEY BACK GUARANTEE*



NEW TAPE UTILITY C2

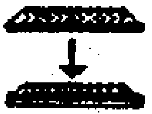
TU.C2 PACKAGE COPIES MOST TAPES EVEN JERKY, COUNTDOWN, + FIREBIR* LOADERS. EVEN CONVERTS MANY SECURED PROGRAMS TO NORMAL SPEED FOR RELIABLE LOADING. MEASURES LOADING SPEED, COPIES EXACTLY BLOCKS - EVEN OVER 51K, + SHORT TONE LEADERS ETC. £8.50

SAM TAPE AND ASSEMBLER

SAM-ASSEMBLER - WE HAVE TAKEN OUR Z80 TOOLKIT ASSEMBLER AND RE-WRITTEN IT WITH MORE FEATURES FULL SCREEN EDITOR, ETC, ETC. YOU MUST HAVE THIS TO WRITE YOUR SAM MACHINE CODE!! £8.99

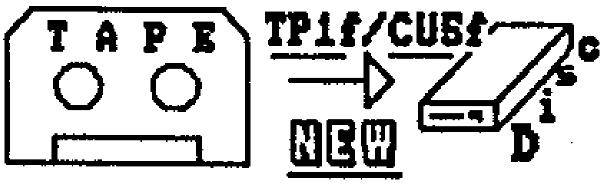
SAMTAPE - EASY TRANSFER OF YOUR SPECCY PROGRAMS TO SAM - PRESS NMI TO STOP - SAVE TO DISK, ENTER POKES, PEEK, ETC. £8.99.

PLUS-3-MATE THE DISC UTILITY AND MICROMATE



+3-MATE HAS 5 DISC UTILITY PROGRAMS INC FULL CAT, DISC REPAIR/VERIFY, LOCK-OUT, SECTOR EDITOR, MULTIPLE ERASE + FILE COPIER, DISC INFO, BOOT PROG, ETC. INCLUDES CLONE - WHOLE DISC COPIER (inc PROTECTED DISCS). ON DISC FOR £12.99 - WHY PAY MORE!

MICROMATE - SIMILAR TO +3MATE FOR MICRODRIVE. FORMATS UP TO APPROX 104K - RECONDITIONS CARTRIDGE + MUCH MORE - £11.99



FOR ALL DRIVES (inc M/DRIVE, +3, +D, ETC) - MANAGES SOME PROGRAMS HARDWARE DEVICES CAN'T DO.

1. CU5f - DOES HUNDREDS OF PROGRAMS EASILY INCL PULSED, COUNTDOWN LOADERS + FIREBIR* - ABOUT 50% OF SOFTWARE. £7.95

2. TRANSPACK 1f - OUR UNIQUE COMPLETE TRANSFER PACKAGE (INC 128K) - INCLUDES CU5f PLUS OTHER SOFTWARE AND SAMPLE TRANSFERS FOR PROGRAMS CU5f CAN'T MANAGE. NOT FOR THE NOVICE. £11.99

3. INFORMATION BOOKS 1, 2, OR 3. EACH HAS AT LEAST 160 TRANSFERS TO USE WITH TRANSPACK 1. SAE FOR LISTS. £2.50 PER BOOK.

WE ARE THE ONLY COMPANY OFFERING A COMPLETE TAPE TO DRIVE PACKAGE

Z80 TOOLKIT 2

"EXCELLENT VALUE FOR MONEY" "BRILLIANT PACKAGE" "IF ITS NOT IN TOOLKIT YOU DON'T NEED IT". YS JULY 1988.

OUR SUPERB PROGRAM FOR ALL WHO USE MACHINE CODE - UNRIVALLED IN PRICE AND VALUE FOR MONEY. PACKAGE INCLUDES SINGLE STEP - HELPS YOU UNDERSTAND AND DEBUG M/CODE. POWERFUL ASSEMBLER WITH EASY TO USE FULL PAGE EDITOR, AND A PROGRAM TO DO THE REVERSE - M/CODE TO SOURCE. TOOLKIT WITH DISASSEMBLER, HEX TO DEC, LOADER MODE, PRINTER OUTPUTS ETC. TRANSFERS TO DISC. ONLY £8.99. AMSTRAD CPC VERSION £9.99.

STATE SPECTRUM TYPE WHEN ORDERING. (e.g 128K AND +D). * ONLY ON SOFTWARE

+3 DISKS FOR £2.30 MIN 4 - POST FREE.

UK POST FREE, +£1 EUROPE, +£2 REST

CODE COMPRESSOR AMAZING AT £4.50!

LERM, DEPT YS, 11 BEACONSFIELD CLOSE, WHITLEY BAY
TYNE+WEAR. NE25 9UW. (091)2533615 PROP:L.Evans.

NEV'S HELP PAGE

By:- Nev Young.

J.Murphy of Mitcham has noticed a little bug in the PLUS D. That is that some times when he does a CAT he gets a ? just as if the DOS had not been loaded. But pressing ENTER a second time and it works fine. The DISCiPLE also does this and it is caused by the shadow ROM getting its flags out of sync. Each time you enter any command that uses the shadow ROM a byte within the DOS is set to say that you have entered the ROM. This allows the ROM to make calls to the Spectrum ROM and in the event of an error the PLUS D now knows it is a real error and not just a syntax failure. Without this the PLUS D would try and interpret the line all over again and get itself into an infinite loop. When the PLUS D has finished and is returning to the Spectrum ROM it resets the flag byte.

Unfortunately there are one or two occasions when the PLUS D makes a call to the Spectrum ROM but NEVER returns. This means that the next time the shadow ROM is paged in it thinks there has been an error. This causes control to be passed back to the Spectrum error handler and you get the ?. But the flag has now been reset so try again and it works.

Fortunately this only happens when entering commands directly and not when running a program, so it is only a minor annoyance.

Your other problem of the PLUS D locking up the keyboard is a well known one and a fix was published in Format issue 2/10.

I have received over a dozen letters about DFLIP since it was published in issue 3/6. I hope you have all got it working now that the printing errors have been corrected. (A note for any hopefuls trying to get an article in. I sent that in 14 months before it got published and had quite forgotten about it).

Of all the letters one from John Littler in Kent caught my eye. He uses Gens 4 for an assembler whereas I use the OCP editor/ assembler. There would appear to be a difference between the two. The source for DFLIP did not have an ORG at the start. Now OCP assembled this starting at address 0. John tells me that line 330 should have been LD HL with the address of D1 minus address BEGIN plus 7126. This tells me that Gens doesn't assemble from address 0. So to be consistent there should have been an ORG 0 on line 130.

Now the part that caught my eye John wonders why line 330 was : LD HL,D1+7126 and not LD HL,7189 as even when relocated the expression always gives the value 7189. there are a number of places where I used an expression that always gives a constant ie lines 480, 160, 340, 490, 147. The reasons are really very simple. Firstly using in line constants is not really good programming practice. Secondly I am not as good as the Spectrum at arithmetic, and thirdly only the final program produces the constant values that you have worked out. During development there where lines of code being added removed or changed and to work out the constant values would be a time consuming task. As well as unnecessary. For example take the code on lines 330 to 380

```
330 LD HL,D1 + 7126
340 LD BC,D_END - D1
350 LD DE,(WKSP)
360 PUSH DE
370 LDIR
380 JP (HL)
```

(Note for the sharp eyed. Line 380 should have been RET anyway, like line

530) This gives 14 bytes of code. Now what if I had written:-

```
   360 LDIR
   370 LD HL,(WKSP)
   380 JP (HL)
```

Which would have been better but produces code 16 bytes long. Or,

```
   350 JR MVE1
DELETE LINES 360,370,380 and change
   500 MVE1 LD DE,(WKSP)
```

Now its only 8 bytes long. I think that by now you should see the main reason for using expressions rather than working out the result of those expressions and using the result. You also asked for some addresses in the PLUS D that are the equivalent of those in Interface 1.

Interface 1	PLUS D
STEND 05B7	CEOS 046E
SHERR 0020	DERR 0020
ERR6 01F0	ERROR 04F0
END Not named in my disassembly	
RERR Not named in my disassembly	

Although these addresses are equivalent I can not guarantee that they are the ones you need. Remember the PLUS D has a completly different ROM to the Interface 1 and many of the routines written for the Interface 1 just will not work without a complete rewrite for the PLUS D. You may, however, find something of interest in my article on adding new commands to the DISCiPLE and PLUS D in issues 2/3 2/4 2/5 and see the inside G+DOS by Steve Warr in issues 2/8 2/9 2/10 2/12.

And finally Adrian Wood of Yeovil would like to know why an Interface 1 will not work on one Spectrum but works fine on another. The first machine seems OK as it drives an Alphacom printer. Also why does pin 23b (think you mean 22U) of the edge connector have +12 volts on it when it is labelled -12 volts.

A lot of red herrings here. Firstly the -12 volts is a misprint in the manual which has never been fixed even to the 128k +2. Pin 22U is connected to the collector of one of the transistors in the +12volt generator. It actually has a pulsed 12 volts on it.

Secondly the ALPHACOM printer, like the ZX printer, uses very few of the signals on the edge connector so the fact that it works means very little.

The answer is, probably, the signal on pin 23U (M1). This signal is not used anywhere in the Spectrum but it is used by the Interface 1 and the DISCiPLE / PLUS D. It is generated by the Z80 when it is reading an instruction from memory. So if this signal is active and the address lines have a value of 8 then an error restart is in force. This will cause the shadow rom to be paged in. I think you may have shorted this pin out with the one next to it. Unfortunately this is a very common hardware problem.

The solution is easy, replace the Z80 CPU chip. If you are not VERY GOOD with a VERY FINE soldering iron then get somebody who is to do it for you. The PCB is VERY DELICATE and can be damaged beyond repair with indelicate handling in this area. Z80 chips can be obtained from most electronic shops or from MAPLINS (0702 552911) order number QW00A (Z80A-CPU) £2.15. Mention Format if you call them.

Well thats all for this month. Keep those letters coming. I will answer as many queries as possible but only through the magazine so please do not send me return postage etc. Also it is in your interest to send me as much info as possible such as program listings etc. as contrary to popular belief I do not have a vast library of programs, I only do this in my spare time!

Write ONLY to Nev Young at:-

FORMAT Help Line,
3, Mitchell Place,
Falkirk,
Stirlingshire,
Scotland, FK1 5PJ.

FASTER THAN BASIC

By:- Ian Cull.

Part 3. MIRA Software's Compilers.

The best way to speed up your programs is to spend a long time learning assembly language and write your programs in machine code. The Basic Compiler (see Vol3 No4) is really the opposite extreme - you didn't need to learn anything new before being able to produce faster programs. This time we look at compilers for languages other than Basic - specifically Fortran and Pascal from Mira Software.

The MIRA Compilers are similar; both support both 48K and 128K Spectrums with tape, Sinclair Microdrives, DISCiPLE/PLUS D, Beta and Opus disc systems and even Wafadrives (48K only - Wafadrives don't work on a 128K). The 128K versions also fully support the RamDisc. There are no versions for the Spectrum+3, and the Spectrum+2A is also likely to give problems.

FORTRAN

Fortran is a very old language, originally designed so that scientists could write their own programs to solve problems instead of having to describe them to computer programmers. Fortran is quite easy for Basic programmers to pick up - its layout is similar (apart from needing 6 spaces at the start of every line) and offers named functions & subroutines (but only 6 character names). Fortrans has very powerful maths functions (Fortran is an abbreviation of FORmula TRANslation); the MIRA compiler omits Double-Precision and Complex variables, still offering REAL, INTEGER & LOGICAL. CHARACTER variables are very poorly supported - stick to Basic if you want to process strings.

Maths functions include LOGs (to base e or base 10) normal & hyperbolic trig. functions (SIN, COS, etc) & the Fortran MAX & MIN functions. The Fortran DO loop is very similar to Basics FOR/NEXT loop, and Fortran also offers block, logical & arithmetic IF statements - enough power for most programs. MIRA have also implemented COMMON & EQUIVALENCE, allowing sharing of variables between subroutines.

Spectrum specific routines include CIRCLE, DRAW, PLOT, BEEP and USR. Fortrans input/output functions are also well implemented; there are many powerful facilities for formatting the results of your programs using the FORMAT command. Input/Output channels must, however, be set up from Basic before calling the code.

PASCAL

Pascal is a strictly defined language, designed originally for teaching purposes (by Nicklaus Wirth). It offers many facilities for structuring programs (making them easier to understand); the MIRA Compiler conforms to BS6192 level 1 so is no exception. A Basic programmer new to Pascal has little chance of success without buying additional books. Because of the teaching background there is little chance of programming errors not being spotted during compilation; however, learning where and where not to put semi-colons is a tedious process.

Pascal offers a wide range of variable types including ordinals (integers), reals, characters and sets (a variable containing one or more items from a defined list). Pascal sets are particularly powerful, allowing you to restrict a variables values which aids debugging. Character processing is much better than Fortran - one of MIRAs example programs is a Spelling Checker! MIRA have again implemented Spectrum specific functions including all those available for Fortran. Also TAN, ARCSIN and ARCCOS are implemented, as well as being available in the Pascal definition - this is not clarified in the manual.

Files are again implemented by OPENing them to streams in Basic - however 'internal files' can also be created and destroyed by the compiler (on disc versions only) as the program runs.

USING THE COMPILERS

The MIRA Compilers both have a rather unusual user interface. The program source is listed in a similar way to Basic (i.e. in the top screen) but without line numbers. The cursor keys can be used to move up and down the source 1 or 8 lines at a time, or to move to the top or bottom of the program. Pressing ENTER allows new text to be inserted after the current line (until CAPS-SHIFT 6 is pressed), and E allows entry of a single line before the current one. Pressing B or K mark the beginning and end of a block and 'cut it' out, then P allows the block to be 'Pasted' back as many time as necessary (to copy a block you simply re-paste it before moving the cursor). A block can also be deleted completely.

There is a find function (F specifies the search text and begins the search, N continues through the text) and the source can be SAVEd or LOADed to tape, RamDisc (with the 128K version) or to any available drive (according to the version being run). LOADed text is always merged with what is already in memory - this must therefore first be deleted (using the memorable (!) sequence SYM-SHIFT 7 B SYM-SHIFT 6 CAPS-SHIFT 0). An 'option byte' allows setting of where the SAVE/LOAD commands are directed to (i.e. in a separate operation to the actual SAVE/LOAD).

Compilation of loaded source is begun by pressing the X key; after a successful compilation both the source and the compiler are lost - 128K versions save to RamDisc first and 48K disc versions save to disc. Because of this I cannot recommend the 48K tape-only compilers. In addition, there is no access to Basic in the 48K versions once the compiler has been loaded (except a reset) so if a disc/cartridge is full, a new one cannot be formatted or files erased without losing the current source (though you could always save to tape!).

I found a few bugs in the Fortran Compiler - IMPLICIT declarations (which save time by specifying all variable names starting with a particular letter to be a certain type) do not work, and using LOGICAL variables gives a lot of trouble - PRIMES2 is best written using a LOGICAL array but I could not get it to work. The 128K compiler also crashed when I tried to save a new version of a file to PLUS D disc.

There was also a (simple) error in the loader program which transfers the code to Wafadrive.

The Pascal Compiler suffered from bugs too - whenever I finished running a compiled program on the 128K machine, it crashed. This did not occur on the 48K Wafadrive version.

HOW MUCH FASTER THAN BASIC?

Both Fortran and Pascal were designed to be used with Compilers, so they should give good results. Pascal will in general be a little slower since it carries out much better error checking (Fortran would, for example, allow you to overwrite all memory by mistreating an array).

As with all computer languages, the use of integers instead of reals where possible will give great improvement - both languages REQUIRE you to declare every variable as one particular type before using it.

Compilation using the MIRA Compilers is, like Hisoft Basic, two pass. This is particularly noticable on the 48K

Fortran compilers, as the second pass has to load in an additional code block - if an error is found during the second pass then you have to reload everything before fixing it! The Pascal Compiler has both passes in the one program.

I rewrote PRIMES2 (without the displaying of intermediate results) in both Fortran and Pascal to give comparisons with Basic. The Fortran version took an incredible 0.57 seconds to run; this is over 100 times faster than +3 Basic! The Pascal version took 1.5 seconds - indicating the overheads involved in strong error checking.

Because the Compilers automatically save the source code before compilation (except the 48K tape-only versions), this slows compilation even further (especially noticable when I tried the Wafadrive version). This occurs EVEN if you have just saved the source yourself!

SUMMARY

Both the Fortran & Pascal compilers are available from MIRA Software for £15 each - the tape contains all 48K & 128K versions and a few example programs.

However, you may have to budget for additional books since the 'manuals' supplied do not attempt to teach the languages. I cannot seriously recommend using the program on a 48K tape-only Spectrum, and because of the automatic saving of source code & lack of access to Basic the best environment for these Compilers is a 128K Spectrum.

Both compilers are good value and offer Spectrum users the chance to learn alternative programming skills. I really think that MIRA should investigate a +3 and/or +2A version so that all possible Spectrum markets are covered.

Contact MIRA Software at 24 Home Close, Kibworth, Leicestershire LE8 0JT.

PRIMES IN FORTRAN.

```
      PROGRAM PRIMES
C     PRIMES2 rewritten in Fortran
      INTEGER P(1000),PCNT,PTOP,PRIME
      DO 35 I=1,1000
35    P(I)=0
      P(1)=1
      PTOP=1
      PCNT=1
100   IF (PCNT.GT.100) GOTO 200
      IF (P(TOP).NE.0) THEN
         PTOP=PTOP+1
         GOTO 100
      ENDIF
      PRIME=PTOP
      WRITE (2,135) PCNT,PRIME
135   FORMAT(I3,'=',I3)
      PCNT=PCNT+1
      DO 150 I=PRIME,1000,PRIME
150   P(I)=1
      GOTO 100
200   CONTINUE
      END
```

PRIMES IN PASCAL

```
 PROGRAM primes(input,output);
 { PRIMES2 rewritten in Pascal }
 CONST top=100;
 VAR i,j,ptop:INTEGER;
     p:PACKED ARRAY [1..1000] OF BOOLE
AN;
     pcnt:0..top;
 BEGIN
 FOR i:=1 TO 1000 DO p[i]:=false;
 pcnt:=0;
 ptop:=1;
 p[ptop]:=true;
 WHILE pcnt<top DO
    BEGIN
    WHILE p[ptop] DO
       ptop:=ptop+1;
    j:=ptop;
    pcnt:=pcnt+1;
    WRITELN('prime ',pcnt,' is',ptop);
    WHILE j<1000 DO
       BEGIN
       p[j]:=true;
       j:=j+ptop
       END
    END;
 END.
```

* - * - * - * - *

EDITORS NOTE:- It would be nice to see more in FORMAT about languages other than Basic or Machine Code. So come on - lets hear from you.

YOUR LETTERS

Dear Editor,

Any Coupé owner who uses a Spectrum wordprocessor will no doubt be dismayed to find that their wordprocessor, spreadsheet, or database will. not print out data through a disc interface because:-

a) You cannot use the emulator to load up a wordprocessor as the emulator does not contain some of the ROM routines needed.
b) The Speccyrom routine crashes if something is printed to stream #3 as the ZX printer routines in the ROM use a port with the same address as the HMPR (high memory page)' port on the Coupé

So I decided to write a routine to output text through the printer port when using the Spectrum ROM on SAM. I have tested it on The Writer, The Last Word, Artist 2 (even though there is Flash! to use), and Devpac 4 and found that all print without problems.

The routine prints through stream #3 with no characters filtered out, expanded or altered - it is just the minimum required to print through a wordprocessor (or similar package) which doesn't use COPY to do its printing. It is not really designed to use from Basic, but if needed, LPRINT may work but LLIST definitely wont. COPY will cause a crash as it still thinks it's using a ZX printer.

First load a 48k ROM at 65536 (see the Speccyrom article in January's issue. Enter the following pokes:- DPOKE 71102,14446 : POKE 79982,245;1,254,127,237,120,203,71,32,2,207,20,175,211,233,219,233,209,71,32,236,241,211,232,62,1,211,233,201 - and then resave the altered ROM copy. If you saved a ROM from a 128K machine then also do a DPOKE 65611,703 before resaving.

If you want you can enter other pokes before re-saving, like one to turn the BREAK button into a reset back to Spectrum mode by POKE 65638,195,0,0.

Once in Spectrum mode, no alterations are required even after a NEW, RANDOMIZE USR 0 or a reset using the BREAK button.

Note:- pressing the space key while printing will force the computer back to Basic with a RST 8 instruction (useful if the computer hangs up while waiting for an unready printer, but it may crash your wordprocessor.

I hope this routine will be of use to anyone who wants to print before proper SAM software comes along.

Yours sincerely, Daniel Cannon.

Dear Editor,

Could I please pass on a warning to other readers about The Home Computer Club which is based in Swindon. Many people will have seen their leaflets which drop out of many of the glossy magazines with some regularity. Some will even be aware that they have a special devision for Spectrum users.

Well don't fall for their promises of 'A wide choice, with something for everyone'. All they do is games, games and more games. None of the books or utilities that their adverts hint at.

So be warned, unless you are a games freak, don't bother sending off.

Yours sincerely, Paul Milton.

Dear Editor,

I've been a FORMAT subscriber since Issue One and have always found it both interesting and value for money. I read with interest the review of the SAM Coupé in the Jan'90 issue and as soon as my local dealer had one in stock I raided the piggy-bank and went out and bought one.

Wow! The review did not do justice to the speed and graphics. I'm only a Basic programmer but the things I can do with SAM are amazing my friends.

Yours sincerely, Basil Harper.

THE SECRETS OF WORD MANAGER

SPECTRUM MACHINE CODE MADE EASY

Part 12.

By:- Francis Miles.

MULTIPLE PROGRAMMING Part 2.

BASIC to machine code. - The only way to get into machine code operation on the Spectrum is from BASIC by way of the USR function; its operand is a RAM address which the computer proceeds to execute, and the value returned by the function is the 2-byte integer held in the BC register when the machine code reaches a final RET.

There are essentially three ways of using the USR function (minor variations are possible):

- RANDOMIZE USR x (as in the "w driv" program) operates the m/c routine, discarding the output from BC, and then executes the next BASIC statement - though not in the case of "w driv", which deletes this part of the BASIC and, after doing various other things, uses the ROM machine code to re-enter BASIC at line 1.

- PRINT USR x uses m/c to find a value, returns it and prints it out, and then executes the next BASIC statement (not used in "Word Manager").

- LET L=USR x: GOTO L operates the m/c routine and then jumps to a BASIC line determined by the BC returned from m/c. [For subtle reasons, which I won't go into, GOTO USR x doesn't work.] This is the command used by the word processor operating BASIC, and what makes possible the free-standing Microdrive operating commands in lines 210-270 (section 3 in the summary given last month).

In "Word Manager", machine code is always entered in this way from line 200 of the BASIC, and always at the same address, 65018 (FDFA hex). The command at 65018 is 7340 JP ZY (ZY being the start of the "main sequence" of the machine code) it performs various preparatory operations - among others, it fills the whole of the text buffer with spaces, thus deleting the remains of the print configuration BASIC and the "w driv" machine code - then displays the main menu showing memory usage, cursor location and a word count. Note that locations 65019 and 65020 contain the address ZY; it is by playing tricks with this address location, in fact treating it as a system variable, that "Word Manager" can jump in and out of BASIC with such careless abandon.

Machine code to BASIC (and back) - The routine for erasing a text file from Microdrive and then saving the text in memory under the same filename goes like this (although DISCiPLE can be operated by the Microdrive commands, this routine is unnecessary for DISCiPLE; unlike Microdrive, it will overwrite a record with another of the same name):-

```
6300 ;erase/save text
6310 MICES   XOR A
6320         CALL CONF
6330         CALL MICNAM
6340         JP Z,MSM
```

[CONF displays a warning (if A is zero, "Tape record will be erased"), and gives the user a chance to jump out via ABORT back to the main menu. MICNAM prompts input of a filename, which goes into the general-purpose buffer at SBF (see below). If a null filename (ENTER only) is input, the program jumps back to MSM which displays the Microdrive menu.]

```
6350 ;set base address for the
6360 ;BASIC save
6370         LD HL,LO
6380         LD (DCO),HL
```

[LO, the length of the text, is kept

separate from all the other system variables at the beginning of the text buffer, and is saved with the text. Its address is found by the BASIC in system variable DCO, and its value is used by the BASIC save command, with a small increment, to fix the length of code to be saved.]

```
 6390 ;set reentry address
 6400        LD HL,MICS1
 6410        LD (65019),HL
```

[This puts the address MICS1 (see below) in place of ZY in the JP ZY instruction at 65018; so when the BASIC next uses the USR 65018 instruction, m/c operation will be executed from MICS1 instead of the normal main sequence.]

```
 6420 ;return to BASIC line 230
 6430        LD BC,230
 6440        JP M.RET
```

[See M.RET below. Line 230 of the BASIC is GOSUB 280: ERASE "m";d;N$: GOTO 200 The subroutine at 280 PEEKs the values of d (Microdrive number), N$ (the filename) and L0 (which is actually the addressof DCO) from the machine code. GO TO 200 returns execution to machine code at MICS1 through 65018.]

```
 ...
 5910 ;reentry from erase/save
 5920 MICS1  LD HL,ZY
 5930        LD (65019),HL
```

[Restoring the normal execution address to the JP ZY command at 65018.]

```
 5940 ;copy filename to Q$
 5950        LD HL,SBF
 5960        CALL QFILE
```

[All keyboard inputs go into the general-purpose buffer at SBF; the last input was a filename, which is now copied by QFILE to a special store called Q$, from which it will be printed as a footnote in future menu displays.]

```
 5970 ;return to BASIC line 250
 5980        LD BC,250
 5990        JP M.RET
```

[Line 250 of the BASIC is GO SUB 280: SAVE *"m";d;N$ CODE L0,4+FN p(L0): VERIFY *"m";d;N$ CODE: GO TO 200 The subroutine at 280 is the same one described above.]

The jump from the Microdrive routines to get back into BASIC goes to M.RET, which is in the middle of the MSH routine. MSH is otherwise entered from the main menu by pressing ENTER to return to BASIC:-

```
 8420 ;ENTER. return to BASIC
 8430 MSH    XOR A
 8440        LD (IRRP),A
```

[Ensures that the interrupt will be disabled next time machine code is entered.]

```
 8450 ;restore maximum ceiling
 8460        LD HL,ZX
 8470        LD (L1),HL
```

[The "ceiling", the top limit of the text buffer, is lowered for slow printing.]

```
 8480 ;return to BASIC line 9999
 8490 LD BC,9999
```

[This will produce the "OK" report without any action on return to BASIC - unless the user has inserted a line 9999 in the operating BASIC, eg CLS #: LIST. The interrupt and ceiling corrections are deliberately skipped, and the value in BC is a different line number, when return to BASIC is from a machine code Microdrive routine.]

```
 8500 ;all RETs to BASIC from here
 8510 ;restore normal interrupt
 8520 M.RET  PUSH BC
 8530        IM 1
 8540        EI
```

[IM 1 must be enabled on return to BASIC, otherwise the computer won't respond to its keyboard; but it is disabled again almost immediately by the BASIC Microdrive routines. However, when "Word Manager" returns to its main sequence through the JP ZY at 65018, the interrupt will be restored according to the value held in IRRP. So if Microdrive routines are called while "slow print" is operating on the interrupt, printing will stop while the text is loaded or saved, but restart as soon as the program returns to its main menu.]

```
 8550 ;correct HL' and stack pointer
 8560         LD HL,10072
 8570         EXX
 8580         POP BC
```

[The alternate HL register must_always be reset to 10072 (2758 hex) before returning to BASIC if it has or may have been given any other value by the machine code - unless you are a very clever programmer indeed! This is because USR is a function, and the ROM function evaluator (SCANNING at 09467/24FBhex) requires HL' to hold the address in ROM to which all functions must return, otherwise it will crash messily. BC is stacked for EXX, otherwise its value would go into BC'.]

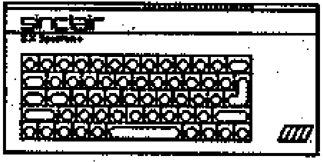
```
 8590         LD SP,(MSSP)
```

[Restoring the stack pointer to where it was when m/c was entered - probably unnecessarily.]

```
 8600         RET ;return to Basic.
```

And that brings me to the end of my series. I hope that, somewhere amongst all these different routines and methods, you will have found something of use to you. Thank you for reading.

Is this the smart-ass computer programmer you were telling Mum about?



0898 299399

MEL CROUCHER-COMPUTER FuN LiNe

New entertaining message~ Starts EVERY MONDAY

STARTS 7th MAY!

Proprietor-B.Everiss, PO, Box 71. Kineton, Warwick, CV35 0XA.

Calls charged at 25p per minute cheap rate and 38p per minute at all other times

BEYOND SIMPLE UDGs

PART 6.

By:- Clyde Bish.

Hello, have you been wandering around our maze we produced last time? You must be quite bored by now so let's start spicing things up with a few doors to open and passages to explore.

First of all load back in your compiled program from last time. (If you missed it you'll have to order a back copy from Bob.) Now add the lines of Program 6a which will provide the subroutines to produce doors that open and close, passage openings with or without pointed arches, holes, ledges and even a crevasse. Oh yes, and torches to light your way.

Having trouble with line 0? This is how you get that one in. Edit line 1 to line 18 to move it out of the way. Now enter:-

```
                  1 LET d=0: RETURN
```

Then type:-

```
POKE (PEEK 23635+256* PEEK 23636)+1,0
```

and press ENTER.

List, and you'll see that line 1 is now line 0! Maybe you can work out how the trick is done - you'll have to refer to the list of System Variables in your manual - but I'll tell you anyway. The System Variable at addresses 23635/23636 is called PROG because it holds the address of the first byte (memory space) of the area in the RAM occupied by your program. The second of these bytes holds the unit of the line number so POKEing this with zero changes the first line number to 0! You'll find you can't erase line 0 in the normal way by just typing the number so its a good place to put your copywrite message as it can't be (easily) moved. (You may wonder why I bothered with a line 0. I have a confession to make. I forgot I needed to put this line first and I wasn't prepared to move all the lines below down one, or make the subsequent changes to GOSUB calls to make room for it. Making mistakes is easy, but to make a real dinger requires skill!)

The reason for the inclusion of this strange line 0 at the very beginning is to provide a RETURN to any GOSUB which uses logic and therefore could, with a false user keypress, produce a call number of zero. Otherwise the program would GOSUB to the original top line (line 1) and produce the wrong illustration.

In program 6a User Defined Graphics (UDGs) are shown as {G P} or {GS 8} which means Graphic Mode and press P or Graphic Mode and press Shift 8. Before we can make use of these new facilities we'll have to amend that and design the necessary UDGs to give perspective to the door opening and flames for the torches. The data for these is in Table A. Enter it using:-

```
FOR F=USR "A" TO USR "P"+7: INPUT A:
POKE F,A: NEXT F
```

reading across the lines. Don't forget to SAVE these UDGs with your program using:-

```
        SAVE "gCODE" CODE USR "a",128
```

If you look at your listing you'll see that it is now littered with strange shapes. To help you in future the end of each line is REMmed with the element it produces. You'll notice the PLOT/DRAW statements for each door element are preceded with a statement to produce a blank space. The reason for this is that you have to make provision for the removal of a door opening before you draw the door, otherwise you get some strange shapes on screen! You'll also notice that a number of lines have bracketed statements within them. Look at line 18 and you'll see what I mean. That mysterious variable d we included last time comes into its own here because it determines, again using the machines logic capabilities, the length , of line that is drawn, depending on whether the ledge required is for a wide room or a narrow passage. You'll see a similar use in line 25 where one wall element is designed for two different uses - part of the wall in a right turn and part of a room wall.

If we're going to make use of these new elements we will have to make alterations in our maze plan. The new layout (with a few extra embellishments we'll come across later) is shown in Fig. 1. For the moment we'll just concern ourselves with the arched opening in room 8 leading to the new room 9 (at the moment without its ledge and hole). Add the lines from Program 6b (some of which can be edited from existing lines) the RUN 80. You'll find that when you reach room 8 it now has an arched opening at its West end, through which you can pass into room 9 (and of course back into room 8 again).

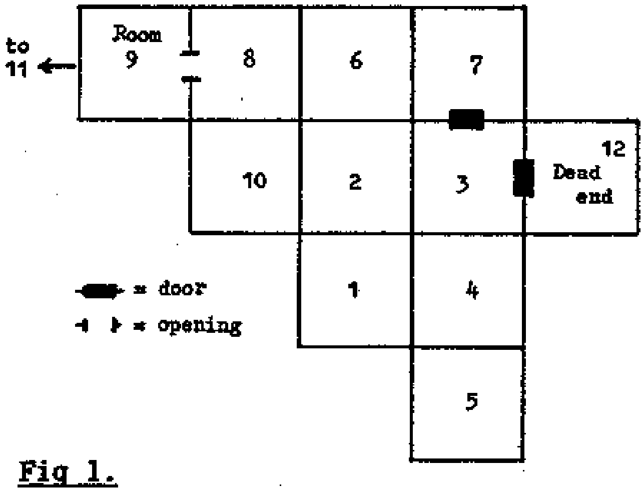


Fig 1.

O.K. Now for the doors. If you examine the new plan you'll see that a door has been inserted between rooms 7 and 3, and also between the latter and a new room 12. Program 7 contains the new lines and necessary amendments (including wall torches for room 3). Type in/Edit your existing program then RUN 80. This time when you move East from room 2 you'll be greeted with a rather different view than previously. In addition to the torches on the wall (the handle colours of which you can change by altering variable i) you'll also see doors East and North. Try turning clockwise through South and West to North and you'll see how the perspective views, including doors and torches, change.

To open a door you must face it, then press "O". Try this with the East door. Now press "C" to close it, then try to go East. You can't as only ghosts, not Adventurers can walk through closed doors! Open it again, then move East. You'll find yourself in a new room 12. Turn and come out but don't close the door yet, I want to demonstrate something else. Instead turn North and open that door, passing through into room 7. Again don't close that door but return to room 3 the long way, i.e. via rooms 6 and 2. When you get back to room 3 you should find both doors still open. You probably don't find this too surprising. After all, that the way you left them! Nevertheless the programming to achieve this is quite close and worth a closer look. The program has to be written to enable a door to be shown open or closed from whatever way you look at it from either side. This is handled by the conditional statements in lines 9005, 9015 and 9016 using information held in array d$() set in line 81. Let's look at the array first. This holds the current door status. The first element is the room number and the second is the orientation (1 = North, 2 = East etc.) Each door is originally set to the closed (c) status. Line 9005 sets the variable o to the orientation number then lines 9015 and 9016 call the open or close_door subroutines. Line 9015 allows you to open a door if (i) the door exists and (ii) it is closed, i.e. if d$(room,orientation) = "c": This is performed by calling the open_door subroutine. (The bracketed statement using the variable r makes sure the door is printed at floor level whether it is in a passage or in a room. The array is then reset to "o". Line 9016 of course works in reverse. The changes within the main program also reflect the restriction of not moving through a closed door. Look at line 301. The conditional statements in brackets will only add 1 to the GOSUB number and so call the open_door illustration as against the closed_door one if they are true. The same applies to the next three lines all of which modify the basic picture produced in line 300. Line 315 makes sure that any door changes that have been made at 9015 or 9016 will affect all views from both sides, then calls the player control subroutine at 9000 to allow other actions to be performed at the same location.

Just one more point on doors and arches. The variable C controls the column print position of the central openings in passages and in rooms. If you alter this you can change the position along the wall of such an opening. It also affects the position of arches. If you do change C remember to reset it back to its original value after printing the arch/opening or you'll have doors which do very stange things when they open!

This open/close doors routine I have been explaining is perhaps the most difficult in the program to understand - it certainly was the most difficult to write! It is however essential that you appreciate how it operates if you are to use the technique in your own programs.

I think I'd better leave you now to let your brain cool down! In the next part we'll concentrate on programming in the sort of things that turns a maze into an adventure. See you soon.

PROGRAM 6a.

```
  0 LET D=0: RETURN
 18 FOR N=112 TO 113: PLOT 0,N: DRAW
    32+(32 AND D=1),-8-(8 AND D=1): N
    EXT N: LET D=0: RETURN : REM corn
    er ledge L
 19 FOR N=112  TO 113: PLOT 255,N: DR
    AW -32-(32 AND D=1),-8-(8 AND D=1
    ): NEXT N: LET D=0: RETURN : REM
    corner ledge R
 32 LET P=2: GOSUB 55: PRINT AT 14,5;
    "O": PLOT 48,88: DRAW -32,16: DRA
    W 0,-96: DRAW 32,16: DRAW 0,64: R
    ETURN : REM left door
 33 PRINT AT 9,2;"{G A}{G B}";AT 10,2
    ;"{GS 8}{GS 8}{G A}{G B}";AT 19,2
    ;"{GS 8}{GS 8}{G C}{G D}";AT 20,2
    ;"{G C}{G D}": FOR N=11 TO 18: PR
    INT AT N,2;"{GS 8}{GS 8}{GS 8}{GS
     8}": NEXT N: RETURN : REM left o
    pening
 34 PRINT AT 9,2;"{G E}{G F}{G P}";AT
     10,2;"{GS 8}{GS 8}{GS 8}{G P}":
    RETURN : REM left arch
 35 LET P=26: GOSUB 55: PRINT AT 14,2
    9;"O": PLOT 207,24: DRAW 32,-16:
    DRAW 0,96: DRAW -32,-16: DRAW 0,-
    64: RETURN : REM right door
 36 PRINT AT 9,28;"{G E}{G F}";AT 10,
    26;"{G E}{G F}{GS 8}{GS 8}";AT 19
    ,26;"{G G}{G H}{GS 8}{GS 8}";AT 2
    0,28;"{G G}{G H}": FOR N=11 TO 18
    : PRINT AT N,26;"{GS 8}{GS 8}{GS
    8}{GS 8}": NEXT N: RETURN : REM r
    ight opening
 37 PRINT AT 9,27;"{G O}{G A}{G B}";A
    T 10,26;"{G O}{GS 8}{GS 8}{GS 8}"
    : RETURN : REM right arch
 38 FOR N=9 TO 17: PRINT AT N,14;"
     ": NEXT N: PRINT AT 13,17;"o": P
    LOT 111,32: DRAW 33,0: DRAW 0,72:
     DRAW -33,0: DRAW 0,-72: RETURN :
     REM centre passage door
 39 FOR N=9 TO 17: PRINT AT N,C;"{GS
    8}{GS 8}{GS 8}{GS 8}": NEXT N: RE
    TURN : REM centre passage opening
 40 PRINT AT 7,C;" {G O}{G P}";AT 8,C
    ;"{G O}{GS 8}{GS 8}{G P}": RETURN
     : REM  centre passage high arch
 41 PRINT AT 8,C;"{G E}{G F}{G A}{G B
    }": RETURN : REM centre passage l
    ow arch (centre)
 42 FOR N=11 TO 19: PRINT AT N,14;"
      ": NEXT N: PRINT AT 15,17;"o":
    PLOT 111,16: DRAW 33,0: DRAW 0,72
    : DRAW -33,0: DRAW 0,-72: RETURN
    : REM centre room door
 43 FOR N=11 TO 19: PRINT AT N,C;"{GS
     8}{GS 8}{GS 8}{GS 8}": NEXT N: R
    ETURN : REM centre room opening
 44 PRINT AT 9,C;" {G O}{G P}";AT 10,
    C;"{G O}{GS 8}{GS 8}{G P}": RETUR
    N : REM centre room high arch
 45 PRINT AT 10,C;"{G E}{G F}{G A}{G
    B}": RETURN : REM centre room low
     arch
 46 FOR N=7 TO 9: PRINT AT N,C;"{GS 8
    }{GS 8}": NEXT N: RETURN : REM ho
    le
 47 PRINT AT 6,C;"{G O}{G P}": RETURN
     : REM  hole arch
 48 FOR N=96 TO 97: PLOT 28,N: DRAW 2
```

```
     00,0: NEXT N: RETURN : REM room 1
     edge
  49 FOR N=96 TO 97: PLOT 60,N: DRAW 1
     36,0: NEXT N: RETURN : REM passag
     e ledge
  50 FOR F=18 TO 20: PRINT AT F,8;"{GS
      8}{GS 8}{GS 8}{GS 8}{GS 8}{GS 8}
     {GS 8}{GS 8}{GS 8}{GS 8}{GS 8}{GS
      8}{GS 8}{GS 8}{GS 8}{GS 8}": NEX
     T F: PLOT 63,32: DRAW OVER 1;-48,
     -24: DRAW 224,0: DRAW OVER 1;-46,
     23: RETURN : REM crevass
  51 PRINT AT 4,1;BRIGHT 1;INK 6;"{G K
     }{G L}";AT 5,1;"{G M}{G N}";AT 6,
     1;INK I;"{G 7}{GS4}";AT 7,1;"{G 5
     }{GS 5}": RETURN : REM near left
     torch
  52 PRINT AT 4,29;INK 6;BRIGHT 1;"{G
     K}{G L}";AT 5,29;"{G M}{G N}";AT
     6,29;INK I;"{G 7}{GS4}";AT 7,29;"
     {G 5}{GS 5}": RETURN : REM near r
     ight torch
  53 PRINT AT 11,10;BRIGHT 1;INK 6;"{G
      I}";AT 12,10;INK I;"{G J}": RETU
     RN : REM far left torch
  54 PRINT AT 11,21;BRIGHT 1;INK 6;"{G
      I}";AT 12,21;INK I;"{G J}": RETU
     RN : REM far right torch
  55 FOR N=9 TO 20: PRINT AT N,p;"
     ": NEXT N: RETURN
```

TABLE A.

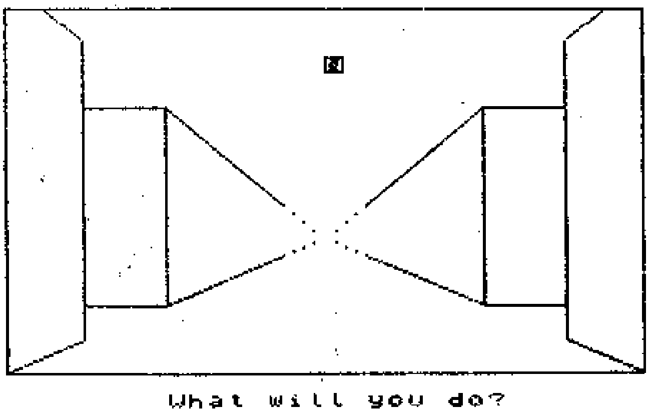
```
240, 252, 255, 255, 255, 255, 255, 255
  0,   0,   0, 192, 240, 252, 255, 255
255, 255, 255, 255, 255, 252, 240, 192
255, 252, 240, 192,   0,   0,   0,   0
  0,   0,   0,   1,   7,  31, 127, 255
  7,  31, 127, 255, 255, 255, 255, 255
255,  63,  15,   3,   0,   0,   0,   0
255, 255, 255, 255, 255,  63,  15,   3
 16,  24,  52,  44,  86,  44,  74,  52
255, 255,  60,  60,  60,  60,  60,  60
  1,   5,   5,   5,  10,  42,  42,  84
  0,  64,  64,  80,  80, 168, 168, 148
 85,  41,  40,  42,  82,  82,  85,  53
 84,  82,  82,  74, 170, 146,  84,  92
  1,   3,   7,  15,  31,  63, 127, 255
128, 192, 224, 240, 248, 252, 254, 255
  0, 250,  33,   1, 243,  54,   0,  62
140,  50,  12, 243,  33,   0, 243, 203
158, 201, 205,  80, 250,  17,   0, 208
235,   1, 128,   5, 237, 176, 201, 205
 65, 236, 220, 106, 244, 201,  60,   0
```

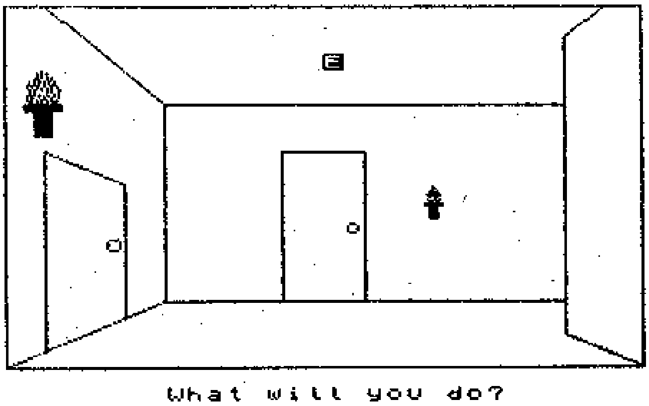
PROGRAM 6b.

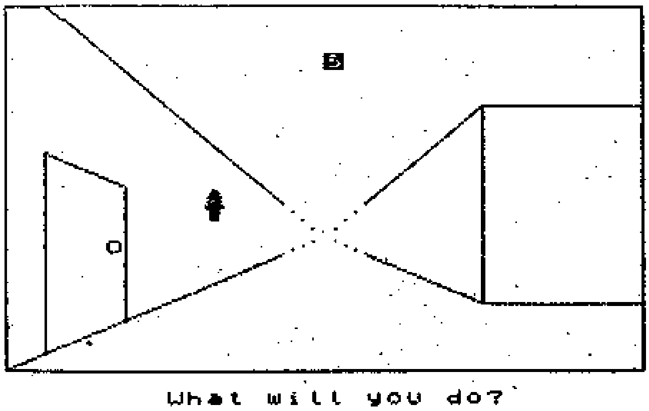
```
  80 BORDER 6: INK 7: PAPER 0: CLS
 801 IF O$="N" THEN GOSUB 33: GOSUB 34
 802 IF O$="S" THEN GOSUB 36: GOSUB 37
 803 IF O$="W" THEN GOSUB 39: GOSUB 40
 830 IF R$="M" THEN GOTO 810+(-210 AND
      O$="E")+(190 AND O$="S")+(90 AND
      O$="W")
 900 GOSUB (17 AND O$="N")+(10 AND O$=
     "E")+(16 AND O$="S")+(10 AND O$="
     W")
 901 IF O$="E" THEN GOSUB 43: GOSUB 44
 910 LET L=9: GOSUB W
 920 IF R$<>"M" THEN GOTO 900
 930 IF R$="M" THEN GOTO 910+(-110 AND
      O$="E")+(190 AND O$="W")
```

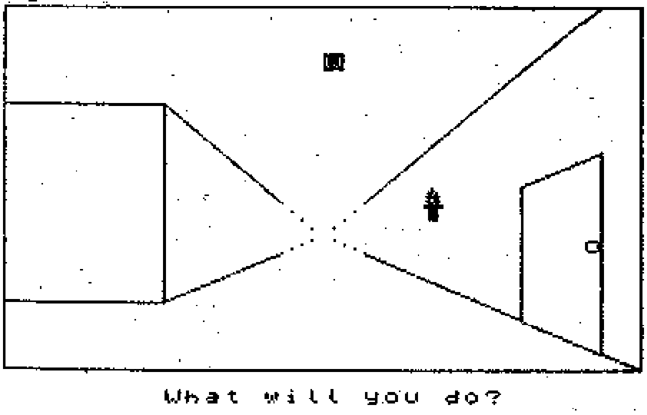
PROGRAM 7.

```
  81 DIM D$(12,4): LET D$(3,1)="c": LE
     T D$(7,3)="c": LET D$(3,2)="c": L
     ET D$(12,4)="c"
  99 LET I=2: LET R=0: LET D=0: LET C=
     14: LET O$="N": LET W=9000
 301 IF O$="N" THEN GOSUB 35+(D$(3,2)=
     "o"): GOSUB 38+(D$(3,1)="o"): GOS
     UB 52: GOSUB 53
 302 IF O$="E" THEN GOSUB 32+(D$(3,1)=
     "o"): GOSUB 38+(D$(3,2)="o"): GOS
     UB 51: GOSUB 54
 303 IF O$="S" THEN GOSUB 32+(D$(3,2)=
     "o"): GOSUB 53
 304 IF O$="W" THEN GOSUB 35+(d$(3,1)=
     "o"): GOSUB 54
 315 LET D$(7,3)=D$(3,1): LET D$(12,4)
     =D$(3,2): IF R$="O" OR R$="C" THE
     N GOTO 310
 330 IF R$="M" THEN GOTO 310+(390 AND
     O$="N" AND D$(3,1)="o")+(-110 AND
      O$="W")+(90 AND O$="S")+(890 AND
      O$="E" AND D$(3,2)="o")
 701 IF O$="S" THEN GOSUB 42+(D$(7,3)=
     "o")
 702 IF O$="E" THEN GOSUB 35+(D$(7,3)=
     "o")
 710 LET L=7: LET R=1: GOSUB W
 715 LET D$(3,1)=D$(7,3): IF R$="O" OR
      R$="C" THEN GOTO 710
 730 IF R$="M" THEN GOTO 710+(-110 AND
      O$="W")+(-410 AND O$="S" AND D$(
     7,3)="o")
1200 GOSUB (15 AND O$="N")+(8 AND O$="
     E")+(14 AND O$="S")+(8 AND O$="W"
     )
1201 IF O$="W" THEN GOSUB 38+(D$(12,4)
     ="o")
1210 LET L=12: GOSUB W
1215 LET D$(3,2)=D$(12,4): IF R$="O" O
     R R$="C" THEN GOTO 1210
1220 IF R$<>"M" THEN GOTO 1200
1230 IF R$="M" THEN GOTO 1210+(-910 AN
     D O$="W" AND D$(12,4)="o")
9005 LET O=(1 AND O$="N")+(2 AND O$="E
     ")+(3 AND O$="S")+(4 AND O$="W")
9015 IF R$="O" AND D$(L,O)="c" THEN GO
     SUB 39+(4 AND R=1): LET D$(L,O)="
     o"
9016 IF R$="C" AND D$(L,O)="o" THEN GO
     SUB 38+(4 AND R=1): LET D$(L,O)="
     c"
```


What will you do?


What will you do?


What will you do?


What will you do?

WRITING FOR FORMAT

FORMAT is always on the look-out for articles and programs to publish in these pages. Articles can be on any computer-related subject not just the Spectrum or the SAM Coupé. They can be just half a page, a long series, or any length in between.

Don't worry too much about spelling and things like that (the Editor can't spell either) we will sort things out. Just put it down as clearly as you can.

It is best if you send your article as a word processor file, on disc or tape, but please include a printed copy so we can look at it straight away. Pack any pictures flat or better still include SCREEN$ files so we can print them out here.

We are urgently looking for writers to produce articles on the following subjects:-

COMPUTERS IN EDUCATION

PROGRAM CONVERSIONS TO DISC

HARDWARE PROJECTS

BUSINESS SOFTWARE

We also require REVIEWS of software (no games please) and hardware, both new and old. If you want to do a review please ring first to confirm no-one else is reviewing the same product.

Each month we try to print a mix of articles in FORMAT, this means that some articles may appear in print almost immediately while others may take several months. It's always worth giving us a ring after about 14 days, just to make sure your work arrived.

Send your work to our address on page 3 or give us a ring on 0452-412572 to talk about it.

GET THE MOST FROM YOUR DRIVE FACILITIES. *STEP UP TO*

NOT SO SOFT-WARE

SPECIFICALLY WRITTEN FOR PLUS D & DISCIPLE

FASTWORD PLUS

Here is a great programme that rejuvenates an old favourite. How often when using TASWORD TWO, have you been stuck for the right word? FASTWORD PLUS offers a complete reference book of words, waiting, ready for use at the press of a key!

This is the Plus D and Disciple Disc version of the sucessful program FASTWORD, a THESAURUS for the Spectrum. Features include ● Full alphabetical index of over 6,000 words ● A display of almost 600 screens ● Equivalent to a 120,000 word printed Thesaurus ● Sense identifaction for each word ● Select a word from the index using the first 2 letters and displayed on the screen is an average of 20 synonyms and associated words ● 2 way scrolling index ● Fast response ● Overlays Tasword 2 ● Words can be looked up in the Thesaurus as you are working on a text file in Tasword 2!

Also supplied is a NEW TASWORD 2 BASIC PROGRAM, specially written for Disc drive operation, offering new features such as selective catalogue, Wordcount etc. A CONVERTER program tailors your Tasword 2 code to give any combination of screen colours plus a new highly readable Tasword 2 character set.

Complete with instructions these 4 programs represent amazing value. Supplied on Disc. ONLY **£9.95**

FILE MASTER

At last! A comprehensive DataBase system that is VERY EASY to use. You will never forget the correct keys to press when using FILE MASTER, we guarantee it!

This versatile and powerful program can be used to store virtually any type of data. File Master enables you to create and maintain sophisticated data files, the type normally associated with more expensive computers. For example a file called 'DIARY' could be created. After typing the relevant data, the program can inform you which of your friends have a birthday in the next month, if the TV licence is due and the date of your next dental appointment. Records of money paid into the bank and cheques drawn could also be kept in the diary, and for an up-to-date total you simply press a key! The variety and scope of the files you can create is almost limitless and each file you create can have an infinite number of records (using segmented files).

Send for your FILE MASTER disc today. ONLY **£8.95**

THE ORGANISER

Features include ● Organisation of programs on any disc in any order you wish ● Makes finding programs in the directory much easier ● Selected programs load faster ● Checks for faulty sectors ● Automatic or manual Sector Repair ● Recovers erased files ... PLUS many more features to make file handling and disc 'house-keeping' so much EASIER.

This is what the experts said:-

"The ORGANISER has a very friendly user interface.... it's rare to find a package so easy to use.... brings out the best features in GDOS.... MGT should bundle it with SAM.... value for money...."
SID MARTIN in COMPUTER SHOPPER.

"Beautifully constructed and presented.... at £5 a bargain.... worth £10 or more just for the Disc Doctor aspects alone.... clearly to be recommended...."
Dr. JOHN WASE in POPULAR COMPUTING WEEKLY.

The ORGANISER is an essential program for ALL Plus D and Disciple users. Supplied on Cassette. ONLY **£5**

DISC MANAGER

THE DISC MANAGER is the most powerful program ever written for the DISCIPLE/PLUS D.

Designed to take advantage of Disc Drive ownership, the Manager keeps track of all the programs on all your discs. ● Storage of up to 27,000 records on one Disc, or 79,920 total. Random File Access. ● Name & Number Discs with fast Autonumber and user pre-defined titles features. No typing in of Data. Press a key and Discs are automaticaly added to appropriate catalogue. ● Fastest ever M/Code Search. 2 modes – Search and Load or Search and List all occurances, then select program to load. ● Plus many other unique features. ● Comes with 16 page manual and demonstration catalogues. ● Operates with 48K or 128K Spectrums. Send for the DISC MANAGER today... and you'll soon wonder how you ever Managed without it! NORMALLY £14.95

SPECIAL INDUG PRICE **£12.95**

BETTERBYTES

SPECIAL OFFER!

FREE ORGANISER TAPE
With All Orders
Over £20

10 SPITTAL TERRACE · GOSFORTH
NEWCASTLE UPON TYNE NE3 1UT · TEL: (091) 285 6185

IF OUTSIDE EUROPE SEND £1.50 p&p FOR EACH ORDER

SOFTWARE THAT IS SPECIFICALLY WRITTEN FOR PLUS D & DISCIPLE

SPECIFICALLY WRITTEN FOR PLUS D AND DISCIPLE · SPECIFICALLY WRITTEN FOR PLUS D AND DISCIPLE

SPECIFICALLY WRITTEN FOR PLUS D AND DISCIPLE · SPECIFICALLY WRITTEN FOR PLUS D AND DISCIPLE ·

PLEASE STATE YOUR DRIVE TYPE, SIZE, ETC, INDUG MEMBERSHIP No. AND IF REQUIRED FOR DISCIPLE OR PLUS D